

IBM No. 69-NC7-027

V/STOL INERTIAL NAVIGATION
WITH
RADAR UPDATE CAPABILITY

Prepared Under Contract Number NAS 12-610
by
International Business Machines Corporation

for
Electronics Research Center
National Aeronautics and Space Administration
Cambridge, Massachusetts
August 1969

Reproduced by the
CLEARINGHOUSE
for Federal Scientific & Technical
Information Springfield Va. 22151

N69-37254	
(ACCESSION NUMBER)	(THRU)
141	1
(PAGES)	(CODE)
NASA-CR-86234	02
(NASA CR OR TMX OR AD NUMBER)	(CATEGORY)

V/STOL INERTIAL NAVIGATION
WITH
RADAR UPDATE CAPABILITY

Prepared Under Contract Number NAS 12-610
by
International Business Machines Corporation

for
Electronics Research Center
National Aeronautics and Space Administration
Cambridge, Massachusetts
August 1969

DISTRIBUTION OF THIS REPORT IS PROVIDED IN THE INTEREST OF
INFORMATION EXCHANGE. RESPONSIBILITY FOR THE CONTENTS
RESIDES IN THE AUTHOR OR ORGANIZATION THAT PREPARED IT.

ABSTRACT

In support of NASA ERC's V/STOL Avionics Systems Research Program, IBM derived, programmed, and verified the equations for the Gemini computer which was flown onboard an H-19 helicopter to test radar update of the onboard inertial navigation system. With the primary purpose being to demonstrate all weather landing capability, the radar update is necessary to remove the accrued errors of the onboard system prior to the terminal landing phase. The navigation equations accept accelerometer outputs from the Gemini IMU and compute the vehicle's position and velocity in an earth surface fixed frame. When commanded via telemetry, the computer accepts radar position measurements, filters the data to obtain estimates of position and velocity, and updates the onboard navigation quantities. The resulting flight test results attest to the validity of both the equations and the approach.

CONTENTS

Section	Page
ABSTRACT	ii
SUMMARY	vii
1 INTRODUCTION	1-1
2 SYSTEM FUNCTIONAL DESCRIPTION	2-1
3 NAVIGATION EQUATIONS	3-1
Navigation Frame Selection	3-1
Coordinate Axes	3-5
Equation Implementation	3-7
4 RADAR UPDATE EQUATIONS	4-1
Preprocessor Filter Characteristics	4-2
Preprocessor Filter Implementation	4-11
Mixing Filter	4-15
5 SOFTWARE DEVELOPMENT	5-1
Equation/Program Verification	5-4
6 POST FLIGHT ANALYSIS	6-1
APPENDIXES	
A NAVIGATION EQUATIONS DERIVATION	A-1
B PREPROCESSOR FILTER DERIVATION	B-1
C SYSTEM MATH FLOW	C-1
System Math Flow Symbols	C-32
D MDIU/DCS/DAS QUANTITIES	D-1

ILLUSTRATIONS

Figure		Page
2-1	Basic System Components	2-2
3-1	Coordinate Axes	3-6
3-2	Navigation Equation Implementation	3-8
3-3	IMU Compensation Equations	3-10
3-4	Platform/Gyro Reference Axes	3-11
4-1	Position Estimate Noise Power Reduction vs. Bias Error Due To Constant Acceleration	4-5
4-2	Velocity Estimate Noise Power Reduction vs. Bias Error Due To Constant Acceleration	4-6
4-3	Position Estimate Normalized Autocorrelation	4-8
4-4	Velocity Estimate Normalized Autocorrelation	4-9
4-5	Velocity/Position Estimates Normalized Cross- correlation	4-10
4-6	Preprocessor Filter	4-11
4-7	Optimal Value of Z_0 with Radar Noise as Parameter for Acceleration Error of 0.04 fps^2	4-14
5-1	Simplified Functional Flow	5-3
5-2	System Simulation Implementation	5-5
5-3	Flight Profile	5-7

ILLUSTRATIONS

Figure		Page
5-4	X Channel Velocity Errors	5-8
5-5	Y Channel Velocity Errors	5-9
5-6	Z Channel Velocity Errors	5-10
5-7	Stationary Navigation Simulation Results	5-13
5-8	Effect of Accelerometer Quantization on Velocity	5-14
6-1	Normalized Autocorrelation for Preprocessor ΔX	6-3
6-2	Normalized Autocorrelation for Preprocessor ΔY	6-4
6-3	Normalized Autocorrelation for Preprocessor ΔZ	6-5
6-4	Navigation Velocity After Landing	6-6
6-5	X Channel Stationary Navigation Quantities	6-8
6-6	Y Channel Stationary Navigation Quantities	6-9
6-7	Z Channel Stationary Navigation Quantities	6-10
A-1	Coordinate Axes	A-2
A-2	Gravity Computation Frame	A-9
A-3	Ideal Mechanization	A-9
A-4	Actual Mechanization	A-13
A-5	Phase of Transport Lag	A-13

ILLUSTRATIONS

Figure		Page
A-6	Phase of Forward Rectangular Integration	A-13
B-1	End Point Linear Fit	B-1
B-2	Exponentionally Weighted Least Squares Position Gain	B-8
B-3	Exponentionally Weighted Least Squares Rate Gain	B-9
B-4	Ratio of Standard Deviations of Position Estimate to Measurement ($Z_o = 0.7$)	B-10
B-5	Ratio of Standard Deviations of Velocity Estimate to Measurement ($Z_o = 0.7$)	B-11
B-6	Ratio of Standard Deviations of Position Estimate to Measurement ($Z_o = 0.9$)	B-12
B-7	Ratio of Standard Deviations of Velocity Estimate to Measurement ($Z_o = 0.9$)	B-13
B-8	Error in Position Estimate Due to Constant Acceleration Input ($Z_o = 0.7$)	B-14
B-9	Error in Velocity Estimate Due to Constant Acceleration Input ($Z_o = 0.7$)	B-15
B-10	Error in Position Estimate Due to Constant Acceleration Input ($Z_o = 0.9$)	B-16
B-11	Error in Velocity Estimate Due to Constant Acceleration Input ($Z_o = 0.9$)	B-17

SUMMARY

Under NASA Contract Number NAS 12-610, IBM derived, programmed, and verified the equations implemented in the Gemini computer which was flown onboard the H-19 helicopter during Phase 1B of the Electronic Research Center's V/STOL Avionics Research Program. The equations perform onboard navigation with radar update capability. Navigation is performed in either of two earth-surface-fixed rectangular coordinate frames; the cruise frame with its origin at the liftoff point, or the approach frame with its origin at the touchdown point.

The technique selected for updating the onboard navigation system is based primarily on the digital command system's transmission rate of one data block per second, and the Gemini computer's relatively slow computational speed. The resulting approach utilizes a radar data preprocessor filter which supplies independent samples of position and velocity estimates to the mixing filter at a cycle rate slower than the transmission rate. This allows all available radar data to be utilized in such a manner that the computer speed is not a limiting factor. The mixing filter combines the preprocessor outputs of velocity and position estimates with the onboard information, and updates the navigation position and velocity with the results. Either of two mixing filters may be chosen.

The equations and associated logic were verified via simulation at both the Fortran and Gemini operational program level. These results indicated that the computational errors were at least an order of magnitude less than those due to the IMU. The resulting flight test results attest to the validity of both the equations and the approach.

Section 1

INTRODUCTION

This report describes the work performed by IBM under NASA Contract Number 12-610 in support of Phase 1B of the ERC's V/STOL Avionics Systems Flight Test Program. This phase, which utilizes Gemini hardware, concentrated on the guidance and navigation requirements for an all weather avionics system; with particular emphasis on the critical landing and approach phase. Flight tests of the resulting systems were conducted at Wallops Station, Virginia, where a GSN-5 radar was used to update the inertial system flown onboard the H-19 helicopter. A brief discussion of the basis system components and typical flight sequence is contained in Section 2.

Section 3 describes the navigation equations, their implementation, and the basis for the navigation frame selection. Navigation is performed in either of two earth-surfaced-fixed frames; the cruise frame with its origin at the liftoff point, or the approach frame with its origin at the touchdown point.

The technique selected for updating the onboard navigation system from radar position measurements is discussed in Section 4. To utilize

all available radar data, which is transmitted at one second intervals, a preprocessor filter and a mixing (update) filter are used in such a manner that the Gemini computer's relatively slow computation speed is not a limiting factor. The preprocessor filter, which is a modified version of exponentially weighted least squares, processes all of the radar measurements and supplies independent estimates of position and velocity to the mixing filter at some prechosen multiple of the one second transmission interval.

The mixing filter combines the preprocessor outputs of position and velocity estimates with the existing onboard navigation values of position and velocity. Either of two mixing filters may be chosen; a six/nine state filter with prestored weights, or the "Koenke" filter which replaces the onboard navigation position and velocity values with the estimates, derived from the radar data, during the first update cycle and equally weights the estimates and existing navigation quantities thereafter.

The resulting flight software is described in Section 5, including a typical sequence of events necessary to exercise the computer program. The methods used for program verification are also discussed, including a brief discription of the simulation programs and typical simulation results.

Appendices A and B contain detailed derivations of the navigation equations and preprocessor filter equations respectively. Appendix C

contains the latest Revision B level of the Phase 1B software and includes a math flow symbol list. The flight software input/output quantities and their range and formats are contained in Appendix D.

Section 2

SYSTEM FUNCTIONAL DESCRIPTION

The basic flight test system components as indicated in Figure 2-1 are:

- Inertial Measuring Unit (IMU)
- Onboard Digital Flight Computer
- GSN-5 Radar
- Digital Command System (DCS)
- Digital Acquisition System (DAS)
- Flight Data Recording Equipment

The output pulses from the triad of pulse rebalance accelerometers mounted on the inertial platform are accumulated in buffer storage registers. These are periodically (nominally one second intervals) sampled and cleared by the computer which converts them into increments in velocity. Using these inputs the computer solves the navigation equations to obtain the vehicle's present position and velocity.

Navigation is initiated in the cruise frame (origin at liftoff point), but can be switched to approach frame (origin at touchdown point)

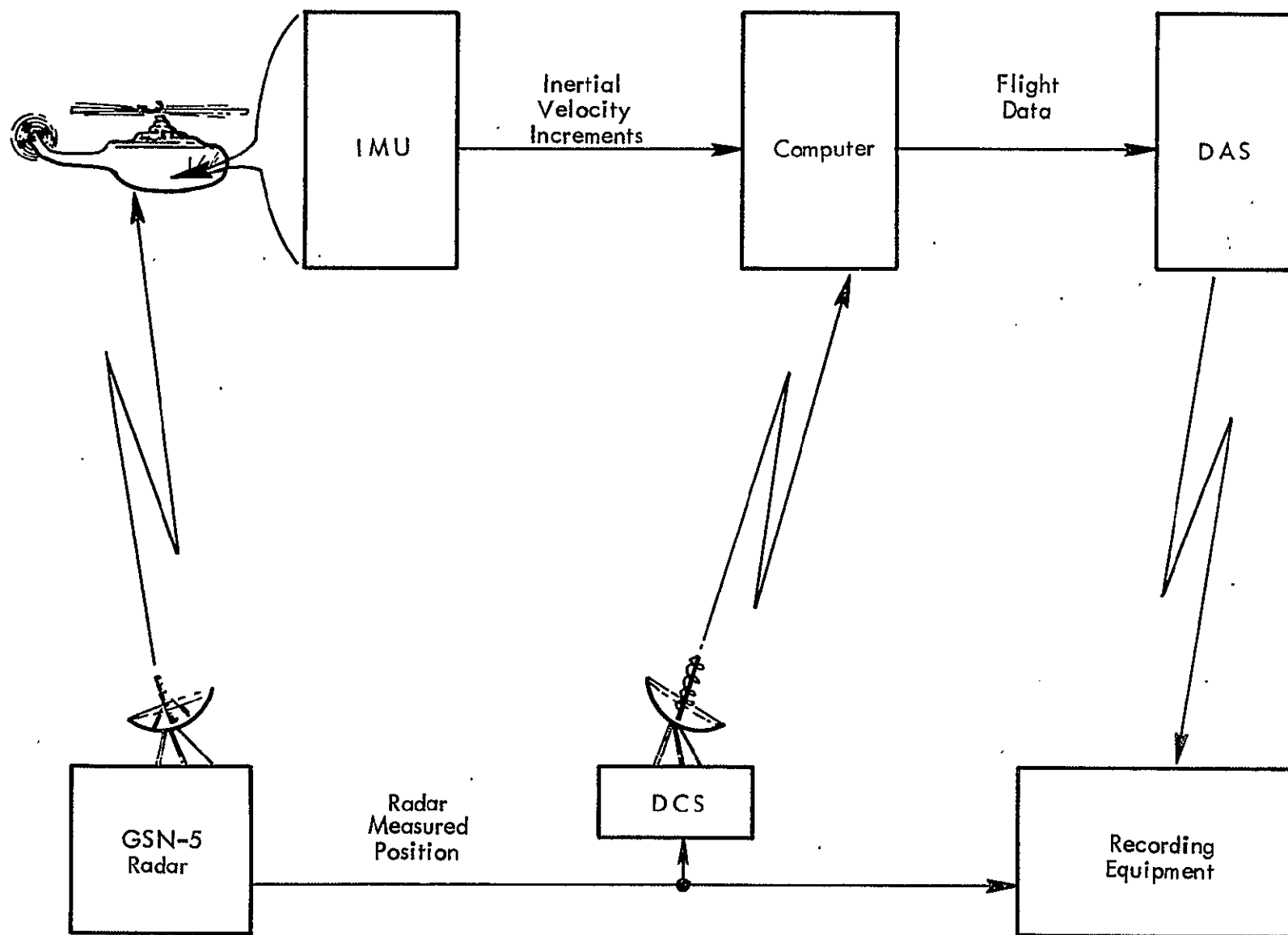


Figure 2-1 Basic System Components

navigation at any time by the proper DCS command. The update mode may also be initiated at any time via DCS command. However, unless the cruise and approach frames are collinear, update must be limited to the approach mode since the radar data is supplied in this frame.

Where the flight plan allows, the GSN-5 radar tracks the vehicle continuously in order that reference data may be recorded for post flight analysis. Twenty-one quantities of onboard flight data are transmitted to ground recorders via the DAS at 2.4 second intervals. The 2.4 second interval is dictated by the Gemini DAS hardware. Since the basic computation cycle is one second, the flight software utilizes buffer storage to assure that all quantities in each transmitted data block pertain to the same flight time.

A typical flight sequence starts with inertial platform alignment with the y axis along the local vertical and the x and z axis nominally East and North, although any azimuth is feasible. After IMU stabilization, and determination of the IMU error correction coefficients, any required changes to the navigation initial conditions and/or correction coefficients stored in the computer are made via the Manual Data Insertion Unit (MDIU).

Navigation is initiated by the computer "start comp" button which also frees the caged inertial platform. At the specified time the switch from cruise to approach frame navigation is initiated by manual DCS command. Another DCS command initiates the update mode, at which time the flight

computer starts processing the radar data. A block of radar data, consisting of time and the three components of position, is transmitted at one second intervals. The computer smoothes several, e. g., ten seconds, of radar data and computes an estimate of position and velocity. Using prestored gains, these estimates are combined with the existing navigation position and velocity to obtain the updated values. The updating continues until a DCS command to return to pure inertial navigation is transmitted. This nominally occurs in the early stage of the landing approach phase.

To obtain a measure of the updated system accuracy, stationary navigation is usually continued for a period of time, e. g., fifteen minutes, after touchdown. This provides an excellent check since the correct solution for velocity and position at the touchdown point is identically zero.

Stationary navigation is also usually performed prior to the actual flight test. By monitoring the navigation outputs, required adjustments to the IMU error correction coefficients can be determined and entered into the flight computer.

Section 3

NAVIGATION EQUATIONS

Navigation is performed in either of two earth-surface-fixed rectangular coordinate frames; the cruise frame with its origin at the liftoff point, or the approach frame with its origin at the touchdown point. Thus, the vehicle's position is supplied with respect to the liftoff point until switchover is commanded, and is thereafter supplied relative to the touchdown point. The approach frame is also defined to be collinear with the radar frame,* thereby simplifying the computation required for radar update of the onboard system prior to touchdown. The choice of the earth-surface-fixed frames has the additional advantage of improved computational accuracy as it allows a more desirable choice of variable scaling for the Gemini computer software.

Navigation Frame Selection

There are several schemes which might be implemented in the V/STOL Phase 1B equations. Three basic schemes (there are several variations) are:

*The GSN-5 radar, which provides data in a cartesian earth-surface-fixed frame, has the capability of offsetting its coordinate frame origin to the desired touchdown point.

1. Navigate in an earth surface fixed (ESF) frame and combine the radar data with the navigation data in this frame.
2. Navigate in an earth centered inertial (ECI) frame and transform the radar data into the ECI frame prior to combining. The combining of IMU and radar data would be in ECI coordinates.
3. Navigate in an ECI frame and transform the position and velocity data into the ESF frame prior to combining with the radar data.

The dominant factors in selecting one of these three basic schemes are equation complexity (e.g., number of transformation required, computer solution time), numerical problems (e.g., scaling, roundoff, truncation), and statistical considerations (e.g., data correlations, noise power reduction). The first scheme listed above and the one selected has the following desirable properties.

1. Reduced scaling problems by virtue that, in the ESF frame, only relative distances and velocities between the V/STOL aircraft and some origin (takeoff or landing sites) appear in the output registers.
2. Reduction in the effect of errors induced by trigonometric subroutines. In the ESF scheme an error in a trigonometric function is roughly equivalent to an error in aligning the platform since trigonometric functions appear only in the differential

equations as rotations of acceleration components. The trigonometric error is on the order of 10^{-6} to 10^{-7} . Thus, the corresponding platform error is one micro "g" or less.

3. Computer induced errors do not propagate through the mixing filters. In the ESF scheme, accumulated error induced by the computations are eliminated at the first radar update since at the first update the filter weights place all of the weight on the radar data and little on the navigation data. Thereafter, the only computer induced errors are those which are equivalent to platform misalignments and these have been shown to be negligible.
4. For the stationary navigation problem, the nominal state of the position and velocity registers must be zero in the ESF frame. For the other two schemes, a nominal trajectory would have to be stored or computed. The effect of computer induced errors could cause non zero conditions which might be difficult to distinguish from other errors. Thus, preflight checkout or alignment is easier to implement in the ESF frame.
5. Since the elevation angle is expected to be less than six degrees and since the basic uncorrelated radar measurements are range, azimuth, and elevation, the correlations among the transformed cartesian ESF radar data (i.e., x, y, z) are also small. Hence,

the digital filter needed to process the radar data from raw position to smoothed position and velocity may be implemented as three two-state uncoupled filters instead of one six-state totally coupled filter.

If the second scheme had been selected, the complexity would have been reduced in the navigation equations (fewer acceleration components to be computed) but increased in the digital filter equations because in order to have the same noise power reduction characteristics a totally coupled filter would have had to be implemented. The radar noise in ECI coordinates cannot be expected to be uncorrelated.

In the translation of radar data into the ECI frame, an error in the trigonometric subroutine can cause computer induced errors on the order of 20 feet. These 20 feet errors are systematic, and digital filtering cannot remove them.

The third basic scheme is the most complex with respect to the navigation equations but has the same digital filter complexity as the first scheme. The increase in complexity in the navigation equations is necessary because both position and velocity information must be transformed from ECI to ESF, i. e., the mixing filter provides for both position and velocity mixing. The numerical error caused by trigonometric subroutines is also present with third scheme. For these reasons the third scheme was discarded.

Time did not permit a complete quantitative tradeoff between schemes one and two. However, it appeared that the first scheme would prove to have fewer numerical problems with no risk in programming due to complexity. The potential complexity problem was circumvented via matrix development of the equations which organized and simplified the bookkeeping problems. The actual scheme mechanized uses two ESF frames. The reason for doing so was primarily esthetic. The navigator initially uses the data in the form of distance and velocity from the liftoff point and later in the flight in the form of distance and velocity to go to the touchdown point. The "y" channel indicates the distance and velocity normal to a line joining the liftoff and touchdown points and so in effect yields "off course" information. The "x" channel indicates the distance traveled or the distance-to-go to touchdown. The z channel does not reflect altitude but height above the tangent plane. This is no loss or compromise because no coordinate frame can provide altitude without a detailed map of terrain fluctuations. However, in the ESF frame the interchanging of z and altitude becomes perfectly valid at the beginning and end of the flight.

Coordinate Axes

The inertial platform and cruise navigation axes are indicated in Figure 3-1. At liftoff ($t = 0$) the origin of the platform frame (X_p, Y_p, Z_p)

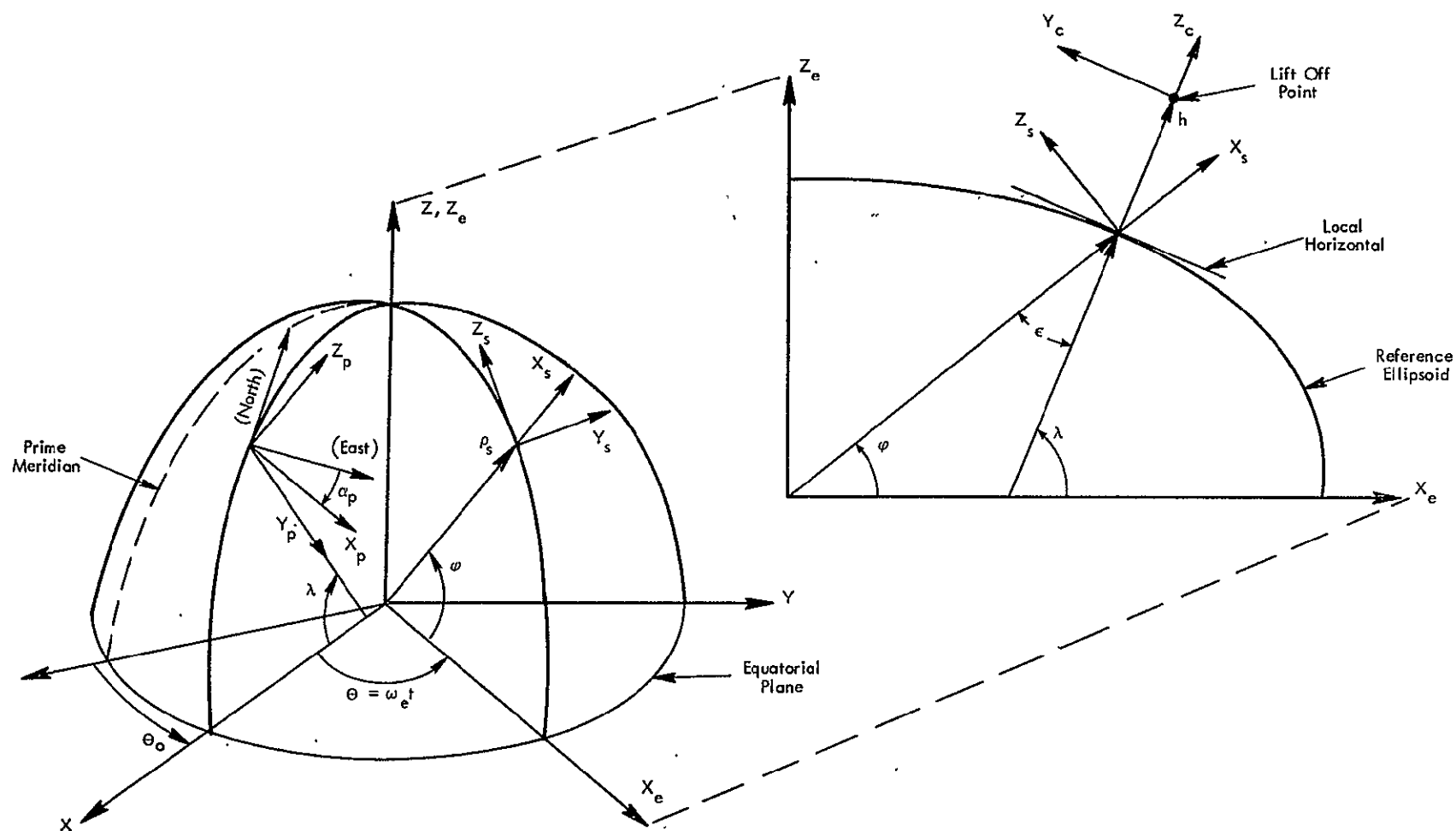


Figure 3-1 Coordinate Axes

and the origin of the navigation frame (X_c , Y_c , Z_c) are coincident. As shown, the platform may be aligned at arbitrary azimuth angles from the nominal North-East orientation. Either or both of the navigation frames may also be aligned at arbitrary azimuth angles with respect to the North-East nominal. In all three cases, a positive azimuth is defined as a positive rotation about the axis along the plumb-bob vertical of the particular frame.

Equation Implementation

A detailed derivation of the navigation equations is presented in Appendix A. The resulting implementation is summarized in Figure 3-2, where the three element vectors are defined as follows:

R_c	Computed position relative to the liftoff or touchdown point.
\dot{R}_c	Computed velocity relative to the liftoff or touchdown point.
$\Delta \dot{R}'_p$	Increments of sensed velocity in navigation frame.
$\Delta \dot{R}_p$	Corrected accelerometer outputs in platform frame.
\ddot{R}_s	Computed acceleration due to coriolis, centrifugal, and gravity.
R_{cp}	Navigation position predicted forward one-half integration interval.

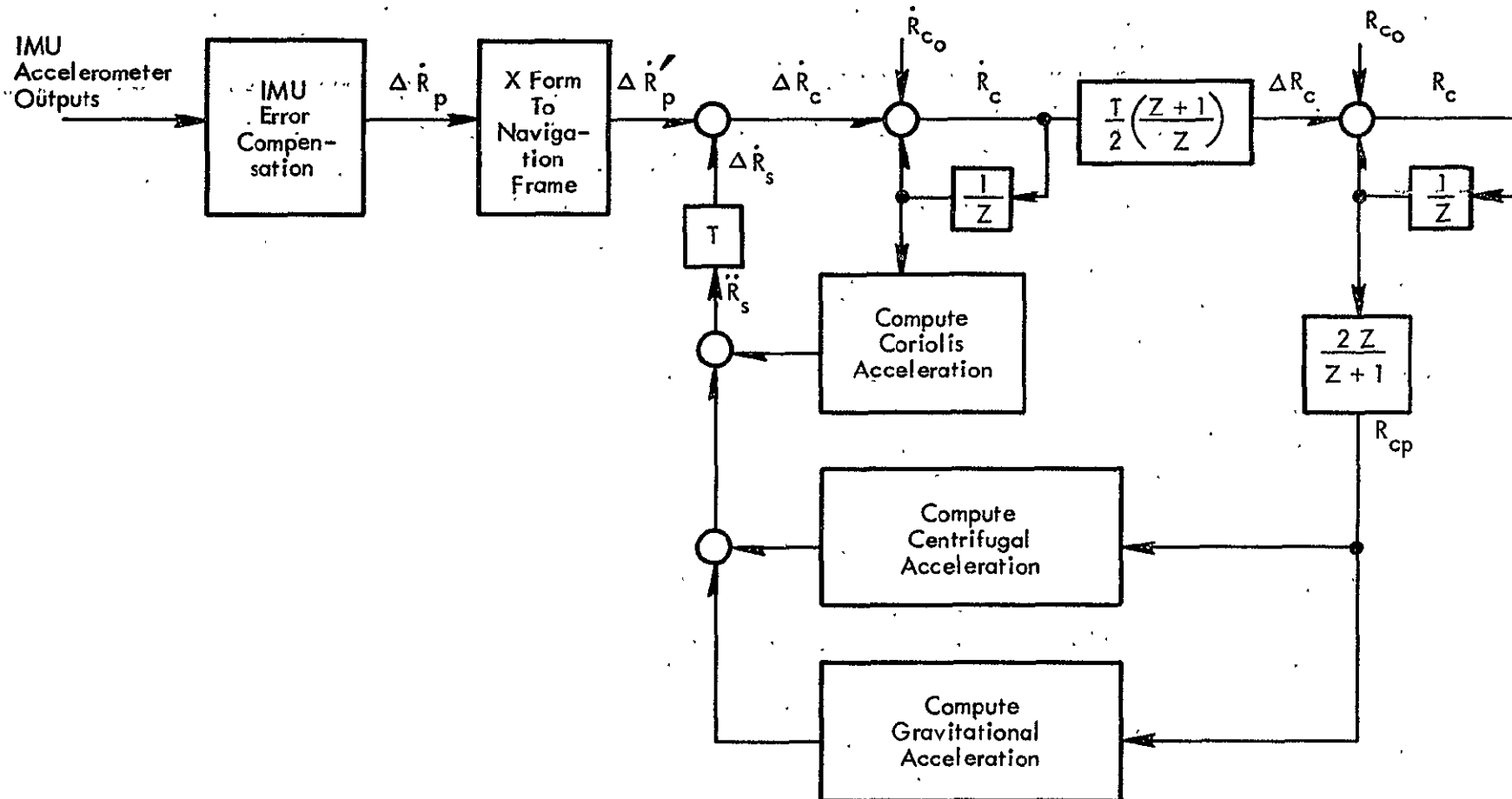


Figure 3-2 · Navigation Equation Implementation

As implied by the above, the same equations are solved for either navigation frame. All that is required to switch from cruise to approach is a reinitialization. The detailed equations are contained in the system math flow in Appendix C.

The navigation equations are solved at one second intervals using trapezoidal integration. As derived in Appendix A, a one-half cycle predictor is used in the position feedback loop to offset the one cycle computational transport lag. This assures the correct phase of the overall position feedback loop.

Since the flight tests are limited to relatively short ranges, an equivalent spherical earth gravity model provides sufficient accuracy. However, in the absence of onboard altimeter information, the magnitude of gravity is held constant, i. e., it is not a function of altitude. This is necessary to prevent the vertical channel instability; and during the critical landing approach phase, after radar update, the resulting gravity error is negligible.

The IMU compensation equations, as contained in the system math flow in Appendix C, are reproduced in Figure 3-3. It shows the computational sequence from accelerometer output (F_{XI}, F_{YI}, F_{ZI}) in quanta to the corrected accelerometer output $(\Delta\dot{X}_p, \Delta\dot{Y}_p, \Delta\dot{Z}_p)$ in feet/second. After correcting for accelerometer bias and misalignment and applying

F_{xI}, F_{yI}, F_{zI} (Accelerometer outputs - Quanta)

$$\begin{aligned} F'_x &= F_{xI} - K_{x4} \Delta t_c \\ F'_y &= F_{yI} - K_{y4} \Delta t_c \\ F'_z &= F_{zI} - K_{z4} \Delta t_c \end{aligned}$$

Accelerometer bias correction

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} K_{x1} & K_{x2} & K_{x3} \\ K_{y1} & K_{y2} & K_{y3} \\ K_{z1} & K_{z2} & K_{z3} \end{bmatrix} \begin{bmatrix} F'_x \\ F'_y \\ F'_z \end{bmatrix}$$

Accelerometer scale factor

and misalignment correction

$$\begin{aligned} \delta_x &= -D_x \Delta t_c - U_{sx} F_x + U_{ix} F_z + \delta_x (-1) \\ \delta_y &= -D_y \Delta t_c - U_{sy} F_y + U_{ly} F_z + \delta_y (-1) \\ \delta_z &= -D_z \Delta t_c - U_{sz} F_z - U_{lz} F_x + \delta_z (-1) \end{aligned}$$

Platform drift

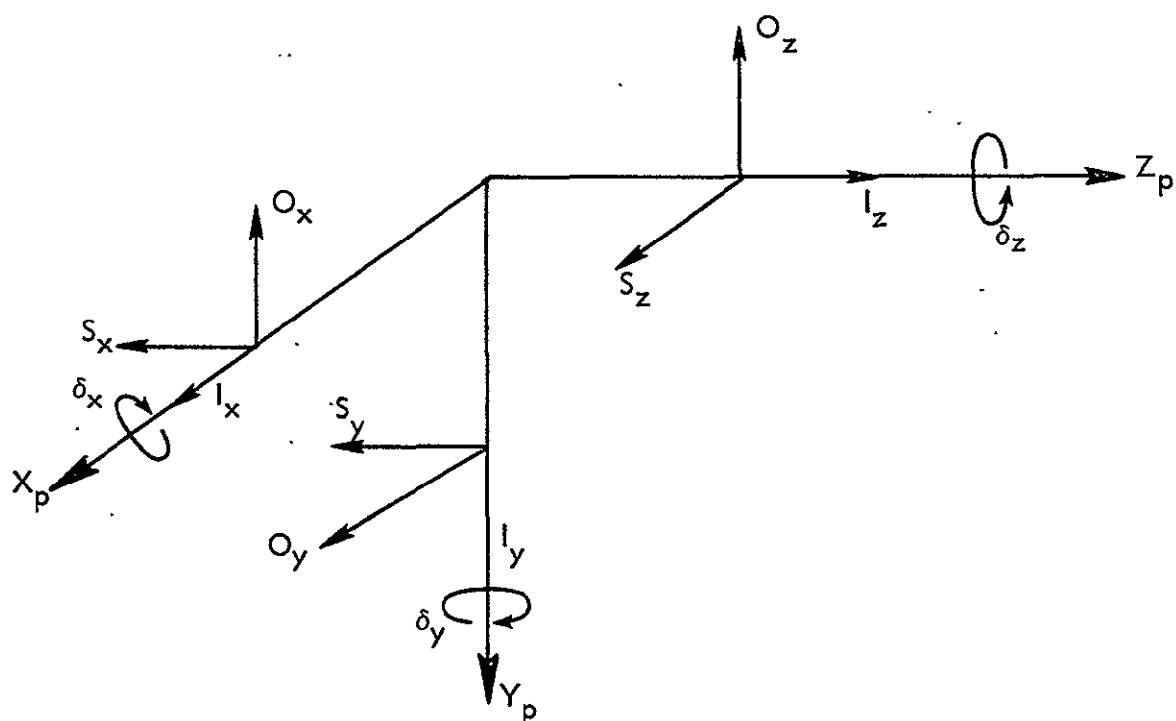
$$\begin{bmatrix} \Delta \dot{X}_p \\ \Delta \dot{Y}_p \\ \Delta \dot{Z}_p \end{bmatrix} = \begin{bmatrix} 1 & -\delta_z & \delta_y \\ \delta_z & 1 & -\delta_x \\ -\delta_y & \delta_x & 1 \end{bmatrix} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}$$

Platform drift correction

$\Delta \dot{X}_p, \Delta \dot{Y}_p, \Delta \dot{Z}_p$ (Corrected accelerometer outputs - FPS)

Figure 3-3 IMU Compensation Equations

the scale factor, the platform drift correction is made using small angle approximations. The sign convention employed is that a positive fixed drift rate (D_x , D_y , D_z) causes a positive gyro gimbal drift which results in a negative platform drift. The same convention applies for the mass unbalance terms with positive input axis accelerations. The orientation of the gyro axes relative to the platform axes is shown in Figure 3-4.



Legend:

- X_p, Y_p, Z_p - Platform Axes
- O - Gyro Output Axis
- S - Gyro Spin Axis
- I - Gyro Input Axis

Figure 3-4 Platform/Gyro Reference Axes

Section 4

RADAR UPDATE EQUATIONS

The technique selected for updating the onboard navigation system is based primarily on the DCS transmission rate and the computational speed of the Gemini computer. The DCS transmits one set of data, consisting of time and the three components of radar measured position, each second. (Between eight and nine hundred milliseconds per data set is actually required.) To obtain the potential accuracy available, the raw radar data should be processed at the transmission rate. At this computational rate, however, the type of filtering that may be employed is limited by the speed of the Gemini computer. For example, to execute a Kalman type filter, where the weighting coefficients are computed in real time, requires considerably greater than the available one second per cycle.

The efficient alternative which was chosen preprocesses the raw radar position data at its transmission rate of once per second. The preprocessor filter supplies essentially statistically independent samples of both position and velocity estimates to the mixing filter at a fixed cycle rate which is slower than the transmission rate. By this approach all

available data is utilized in such a manner that the computer speed is not a limiting factor. The velocity estimates, in effect, compensate for the slower sampling rate.

Preprocessor Filter Characteristics

The preprocessor filter is a modified version of recursive exponentially weighted least squares. A detailed derivation and comparison of characteristics with classic least squares and exponentially weighted least squares are contained in Appendix B. As shown in Appendix B, under steady-state conditions, the preprocessor filter and exponentially weighted least squares have identical characteristics.

The preprocessor filter fits a linear function of time to the measurements. The equations are:

$$(4-1) \quad R_p = \hat{R}_{n-1} + T\hat{R}_{n-1}$$

$$(4-2) \quad \Delta R = R_r - R_p$$

$$(4-3) \quad \hat{R}_n = R_p + K_1 \Delta R$$

$$(4-4) \quad T\hat{R}_n = T\hat{R}_{n-1} + K_2 \Delta R$$

where the three element vectors are:

R_p Predicted position

R_r Latest measurement with inherent noise

\hat{R}_n End point position estimate

$\dot{\hat{R}}_n$ End point velocity estimate

The precomputed filter gains are:

$$(4-5) \quad K_1 = 1 - Z_0^2$$

$$(4-6) \quad K_2 = (1 - Z_0)^2$$

where $Z_0 = e^{-\frac{T}{\tau}}$ where T is the measurement interval and τ is the filter time constant.

Table 4-1 compares pertinent steady-state characteristics of the preprocessor filter to classic least squares as a function of Z_0 and n. Somewhat more informative are the curves in Figure 4-1 and 4-2, where the square root of the ratio of the noise variance of the estimates to the noise variance of the measurements is plotted versus the bias error that would result from a constant acceleration input. The trade-off that must be made between a long memory filter for noise reduction, and a short memory filter to limit bias error growth must obviously be tailored to the system in question.

Although classic least squares is slightly superior in performance, the recursive preprocessor filter has significant advantages from the standpoint of implementation. Since the gains are only a function of Z_0 , its memory length may be adjusted by changing only one quantity thereby much simplifying the programming requirements. This is

FUNCTION	PREPROCESSOR FILTER	CLASSIC LEAST SQUARES
$\frac{\hat{X}}{X_r}(Z)$	$\frac{(1 - Z_o)((1 + Z_o)Z - 2Z_o)Z}{(Z - Z_o)^2}$	$\frac{2}{(N+1)(N+2)} \sum_{k=0}^N ((2N+1) - 3k) Z^{-k}$
$\frac{T\hat{X}}{X_r}(Z)$	$\frac{(1 - Z_o)^2 Z (Z - 1)}{(Z - Z_o)^2}$	$\frac{6}{N(N+1)(N+2)} \sum_{k=0}^N (N - 2k) Z^{-k}$
$\frac{\sigma_{\hat{x}}^2}{\sigma_{x_r}^2}$	$\frac{(1 - Z_o)(1 + 4Z_o + 5Z_o^2)}{(1 + Z_o)^3}$	$\frac{2(2N+1)}{(N+1)(N+2)}$
$\frac{T\sigma_{\hat{x}}^2}{\sigma_{x_r}^2}$	$\frac{2(1 - Z_o)^3}{(1 + Z_o)^3}$	$\frac{12}{N(N+1)(N+2)}$
$\frac{e_{\hat{x}}}{T^2 \ddot{X}_r}$	$\frac{Z_o^2}{(1 - Z_o)^2}$	$\frac{N(N-1)}{12}$
$\frac{e_{\hat{x}}}{T \ddot{X}_r}$	$\frac{(1 + 3Z_o)}{2(1 - Z_o)}$	$\frac{N}{2}$

TABLE 4-1

STEADY-STATE CHARACTERISTICS

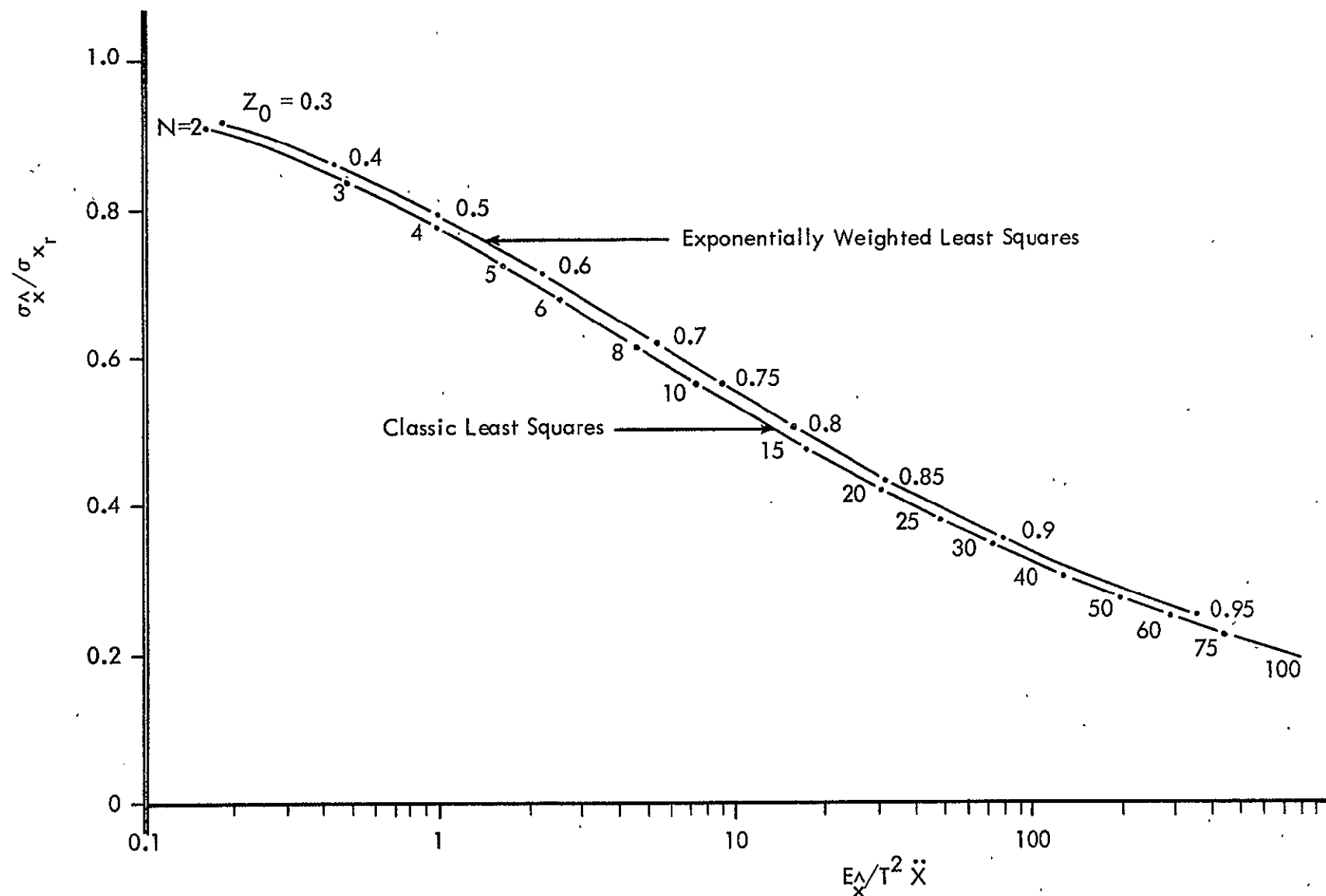


Figure 4-1 Position Estimate Noise Power Reduction vs. Bias Error Due To Constant Acceleration

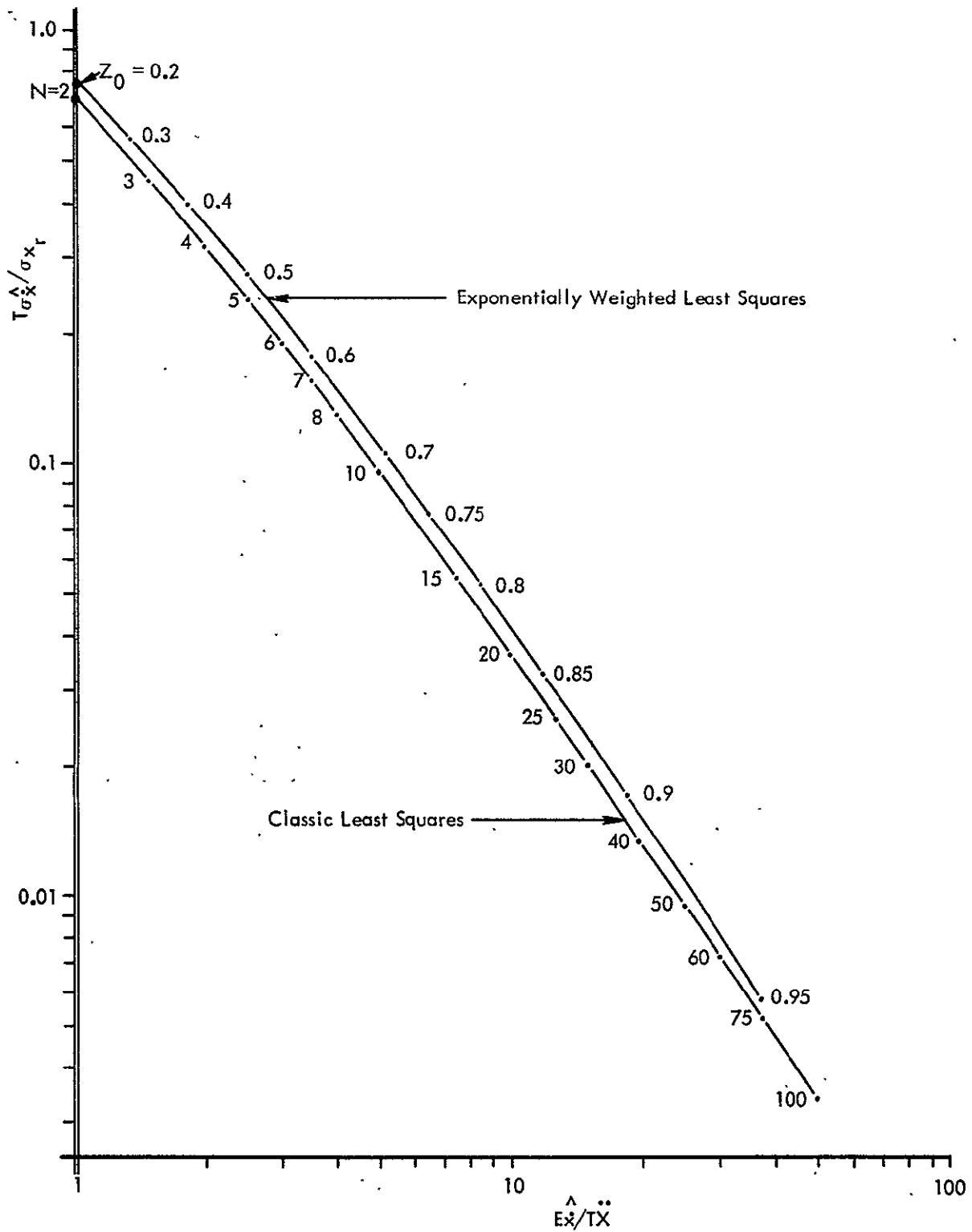


Figure 4-2 Velocity Estimate Noise Power Reduction vs. Bias Error Due To Constant Acceleration

particularly convenient for test programs such as this where the noise models etc. are not well known and changes may be required from flight to flight.

As previously indicated, the primary purpose of the preprocessor filter is to slow the solution rate, from the once per second transmission rate, without discarding any data. However, in order to keep the mixing filter near optimum requires that its inputs (the preprocessor filter outputs at the slower rate) be uncorrelated. The preprocessor autocorrelations and cross correlations for unity power white noise input* are plotted for various values of Z_0 in Figures 4-3, 4-4, and 4-5. The correlation times agree very closely with the number of measurements required for classic least squares as plotted in Figures 4-1 and 4-2.

For example, with $Z_0 = 0.7$, the normalized autocorrelation time for any estimator is less than 0.1 for $nT > 10$. A sliding least squares batch filter with 10 data point memory produces a statistically independent estimate for $nT > 10$. Therefore, either a 10 data point least squares filter or the preprocessor with $Z_0 = 0.7$ produce essentially independent estimates of position and velocity every 10 seconds. Figures 4-1 and 4-2 further indicate that the confidence of the estimates from either filter will be about the same.

* Since the radar servo bandwidth is 8 cps, the radar samples at once per second are uncorrelated.

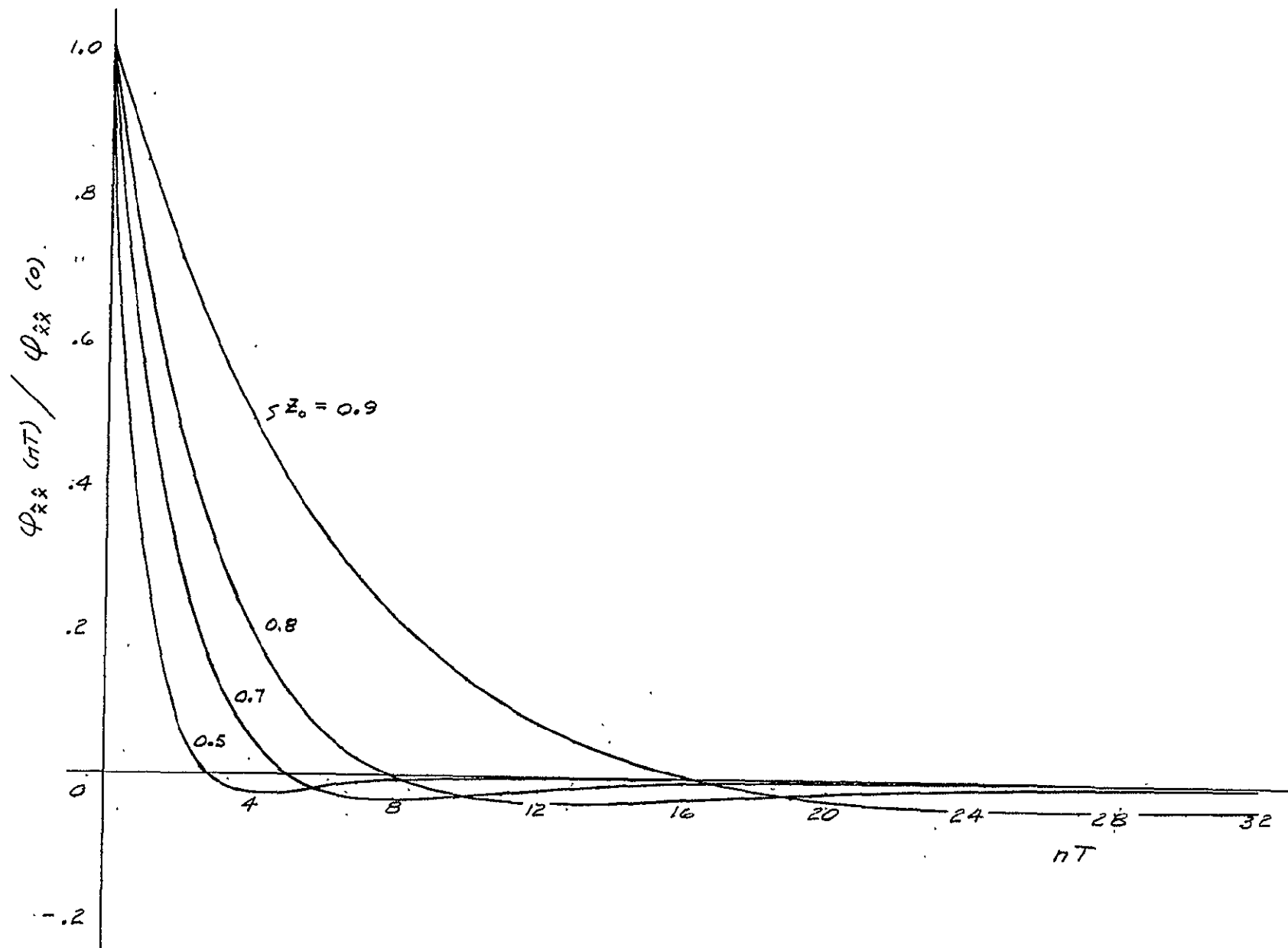


Figure 4-3 Position Estimate Normalized Autocorrelation

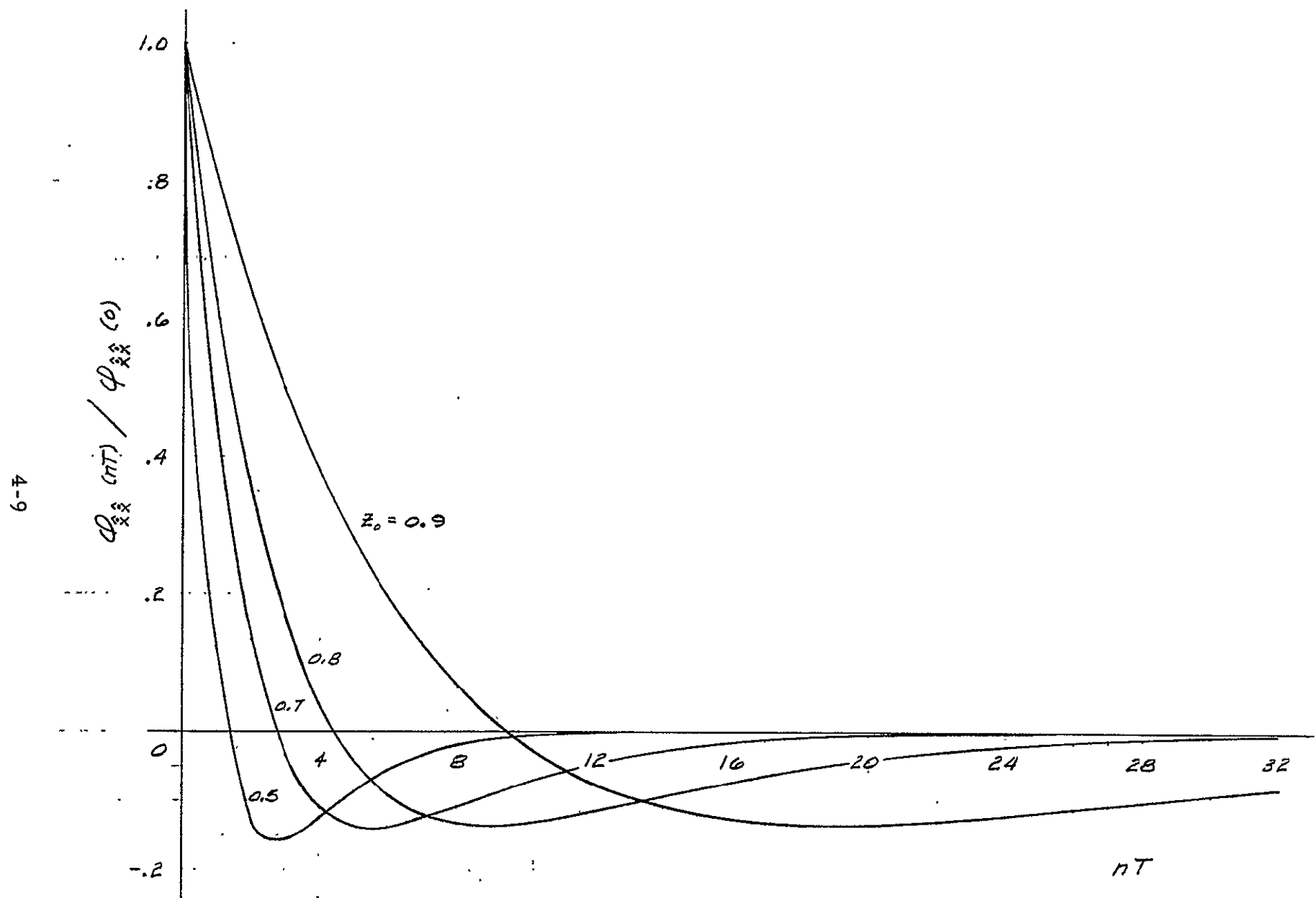


Figure 4-4 Velocity Estimate Normalized Autocorrelation

4-10

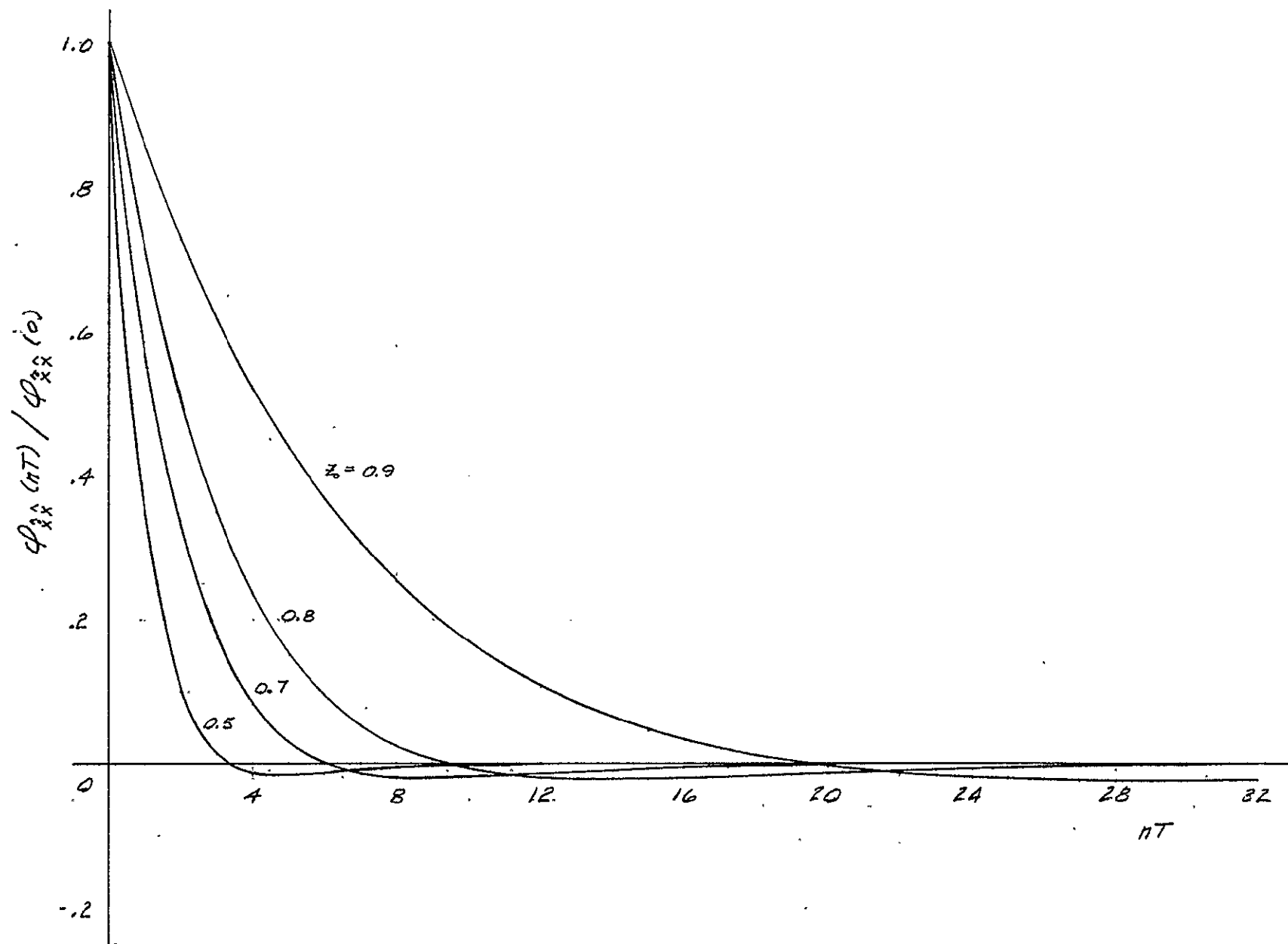


Figure 4-5 Velocity/Position Estimates Normalized Crosscorrelation

From the preceeding, it is obvious that as Z_o is varied to adjust the noise power reduction/bias error growth, the time between updates must be adjusted accordingly to assure that the estimates of position and velocity supplied to the mixing filter are essentially uncorrelated, e. g., for $Z_o = 0.9$, nT should be on the order of 30. This nT corresponds to the MDIU quantity "NTMAX."

Preprocessor Filter Implementation

The preprocessor, as implemented in Phase 1B equations, is shown in block diagram form in Figure 4-6. It differs from equations

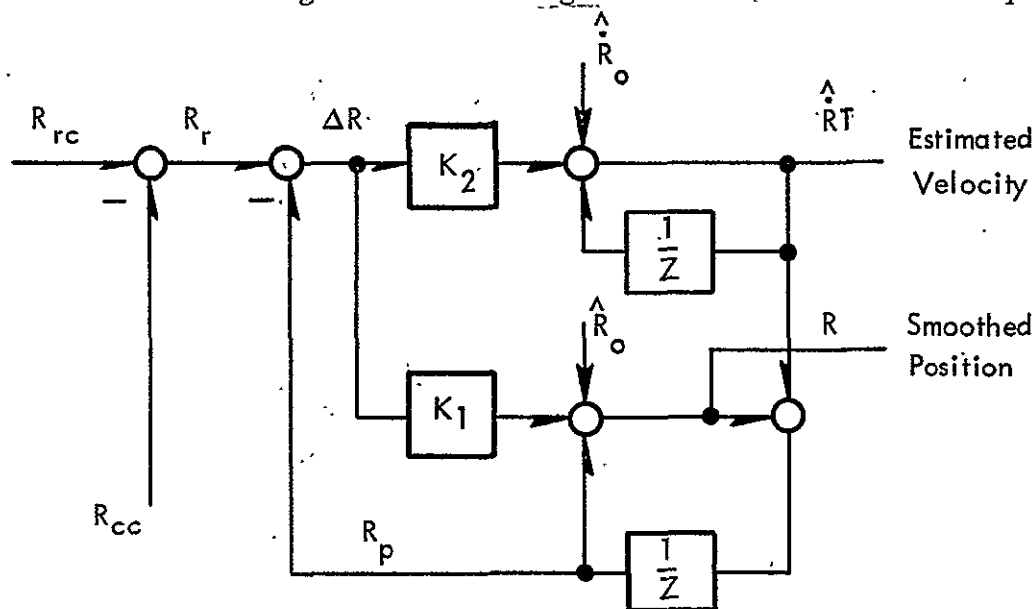


Figure 4-6 Preprocessor Filter

(4-1) through (4-4) only by the R_{cc} term which is necessary for the following reasons. The filter is designed to recognize the average linear motion of the vehicle; thus, the presence of any acceleration must be accounted for. This is accomplished by subtracting the second integral of the total

acceleration (R_{cc}) from the radar data such that, in the presence of no onboard system errors, the difference between the predicted and measured linear motion is due to radar noise.

In equation form this may be described as follows. The radar measurements (R_{rc}) may be modeled as

$$(4-7) \quad R_{rc}(t) = R_o + \dot{R}_o t + \int_0^t \ddot{R}_c(t) dt^2 + \text{noise}(t)$$

where $\ddot{R}_c(t)$ is the vehicle accelerations in the navigation frame and is due primarily to buffeting. Since the preprocessor is designed to recognize linear motion, the buffeting term must be removed from the filter input.

This is accomplished by computing the second integral of total acceleration as seen by the onboard navigation system, i. e., that measured by the IMU plus centripetal, coriolis, and gravity as computed by the navigation equations. This ESF frame acceleration may be expressed as the correct value plus any IMU induced error. Therefore,

$$(4-8) \quad R_{cc}(t) = \int_0^t \ddot{R}_c(t) dt^2 + \int_0^t \Delta \ddot{R}_c(t) dt^2$$

The input to the preprocess filter (R_r) is then

$$(4-9) \quad R_r(t) = R_o + \dot{R}_o t + \text{noise}(t) - \int_0^t \Delta \ddot{R}_c(t) dt^2$$

An approximate value of $\Delta \ddot{R}_c$ can be computed (for rough analysis purposes only) from Equation (4-10) where the transformation matrix is treated as unity.

$$(4-10) \quad \Delta \ddot{R}_c \approx \Delta b + g (\Delta \psi_0 + \Delta \dot{\psi} t)$$

where Δb , $\Delta \psi$, and $\Delta \dot{\psi}$ are the bias, platform misalignment and platform drift rate errors respectively. Using the specified one sigma values of 10^{-4} g's, 30 arc seconds and 0.1 degree/hour, after 2000 seconds of navigation $\Delta \ddot{R}_c \approx 0.04$ feet/second². Using this value of acceleration error, Figure 4-7 gives the optimal value of Z_0 for a one sigma noise of both two feet and ten feet. The optimal value for Z_0 is that which minimizes the sum of the squares of the velocity error due to radar noise and the velocity error due to $\Delta \ddot{R}_c$.

As seen from this figure, the optimal value for Z_0 is 0.825 and 0.7 for a one sigma radar noise of two and ten feet respectively. NTMAX for $Z_0 = 0.7$ should be 10 or greater and 20 or greater for $Z_0 = 0.825$ as determined from Figures 4.1, 4.2, and 4.3. Figure 4-7 further illustrates an important consideration, that too much filtering may actually be detrimental when modeling errors are present.

A second order filter, one which estimates $\Delta \ddot{R}_c$, could be implemented which would do much to eliminate this problem. In fact, this is precisely what IBM recommended during the Phase 1C studies. The penalty for the second order filter is not significant with respect to computational loading but is with respect to the update cycle. As IBM Report Number 69-NC7-024 indicates, to achieve the same confidence in the velocity estimate, as for the first order filter, the data span would have to be increased by approximately a factor of two.

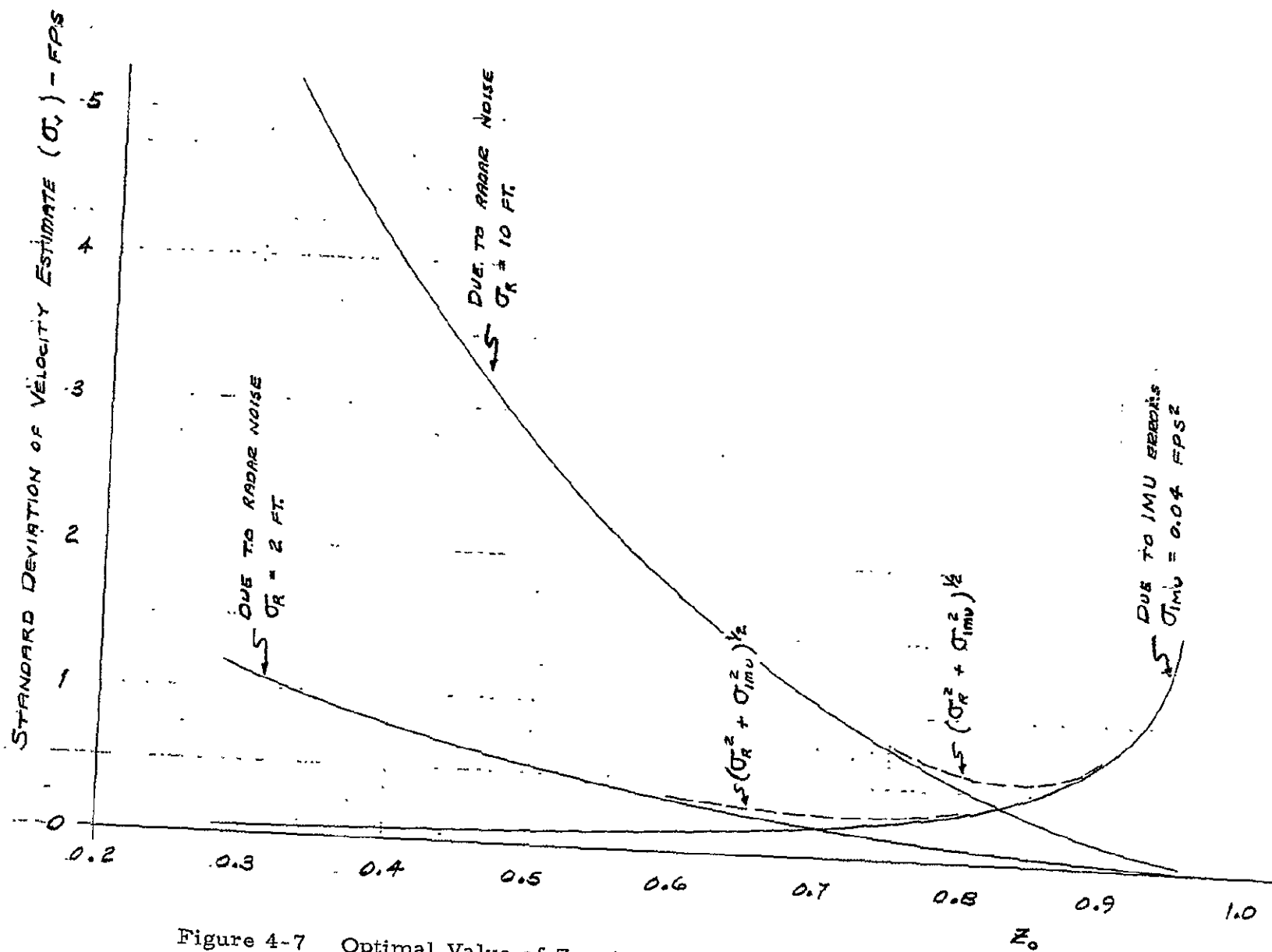


Figure 4-7 Optimal Value of Z_0 with Radar Noise as Parameter
for Acceleration Error of 0.04 fps^2

When the update mode is first initiated, the preprocessor position and velocity should be initialized with sufficient accuracy that an overly large transient does not occur. The radar position data is considerably better than the IMU position data, but the confidence level of the velocity data was a toss-up based on the original radar noise/IMU error models. Therefore, the preprocessor velocity is initialized with the existing navigation velocity, and the position is initialized by a least squares five point fit to the first five radar measurements processed by the computer.

When the measurement noise model is not well known or changes for various flight tests, the filter gains and smoothing times must be altered accordingly. The simplicity of the preprocessor is thus a very desirable feature. For example, if the bias error proves to be small, longer smoothing times (which implies a larger Z_0) may be used to achieve a greater noise power reduction. Such a change requires only two program constants to be changed via the MDIU; i. e., the update interval (NTMAX) and Z_0 . This simplicity and the resulting short computer execution time were of particular significance during the equation development phase when sizing results indicated alternate approaches required excessive computation time on the Gemini computer.

Mixing Filter

The mixing (update) filter combines the estimates of position and velocity, computed by the preprocessor filter, with the existing onboard navigation

values. The filter weights are prestored constants since, as previously stated, the Gemini computer does not have sufficient computational capability for real time solution.

Either of two mixing filters may be chosen, a six/nine state filter or the "Koenke" filter. The six/nine state filter has the option of updating position and velocity only (six state), or to also compute platform drift correction or platform misalignment correction (nine state). The software allows ten sets of gains (540 constants) to be prestored. If updating is allowed to continue beyond ten times, the six state continues to use the tenth gain set over and over. In the case of platform corrections, after the last gain set has been used, the nine state filter is reduced to a six state.

The Koenke filter replaces the onboard navigation quantities with the estimates of position and velocity, obtained from the processed radar data, at the first update; and thereafter, equally weights the estimates and the existing navigation quantities. This filter has advantages besides its inherent simplicity. Since the weights are constant (after the first update) the problem of radar dropouts, and subsequent incorrect weightings, is eliminated. The gains for the Koenke filter and an optimal filter are almost identical for the first two times where the greatest increase in system accuracy is achieved. While the Koenke filter tends to restrict the growth of errors during the update period, the optimal filter tends to

also decrease their rate of growth. However, this decrease may be of little significance; and when problems such as radar dropouts are considered, the optimal approach could actually be inferior.

Section 5

SOFTWARE DEVELOPMENT

The equations discussed in the preceeding sections are presented in system math flow form in Appendix C. This is the final Revision B level of the V/STOL Phase 1B software. The resulting Gemini computer program uses 8,865 of the 12,288 thirteen (13) bit memory locations, and requires approximately 0.7 seconds execution time per cycle with the Koenke mixing filter option. The basic computer computation cycle is executed at one second intervals, the same as the radar data transmission rate. This computation cycle execution is controlled via software as the Gemini computer does not have interrupt hardware.

The input data required by the program is contained in Tables D-1 and D-2 in Appendix D. Those in Table D-1 are entered via the up-link, i. e., the digital command system (DCS), and those in Table D-2 are entered via the manual data insertion unit (MDIU) prior to flight. For those input quantities which require more accuracy then can be obtained through the five character length MDIU, the "write any word" option may be used. This option allows any location in memory to be loaded by three consecutive MDIU operations which provides the full 25 bit data word accuracy.

The simplified flow diagram in Figure 5-1 summarizes program execution of the major events. A typical sequence of actions required to exercise the software are:

Prior to liftoff:

- Computer power on.
- Set mode switch to preflight.
- Load and/or verify input data via MDIU.
- Set mode switch to flight - this initiates cruise frame navigation initialization.
- Press start comp button - Upon recognition the computer turns on computer running light, frees platform and begins navigation.

After liftoff:

- Set LCAN negative via DCS - this initiates approach frame navigation.
- Set LCUD negative via DCS - this initiates execution of the up-date equations.
- Set LCUD positive via DCS - this terminates updating and the system continues inertial navigation in the approach frame.

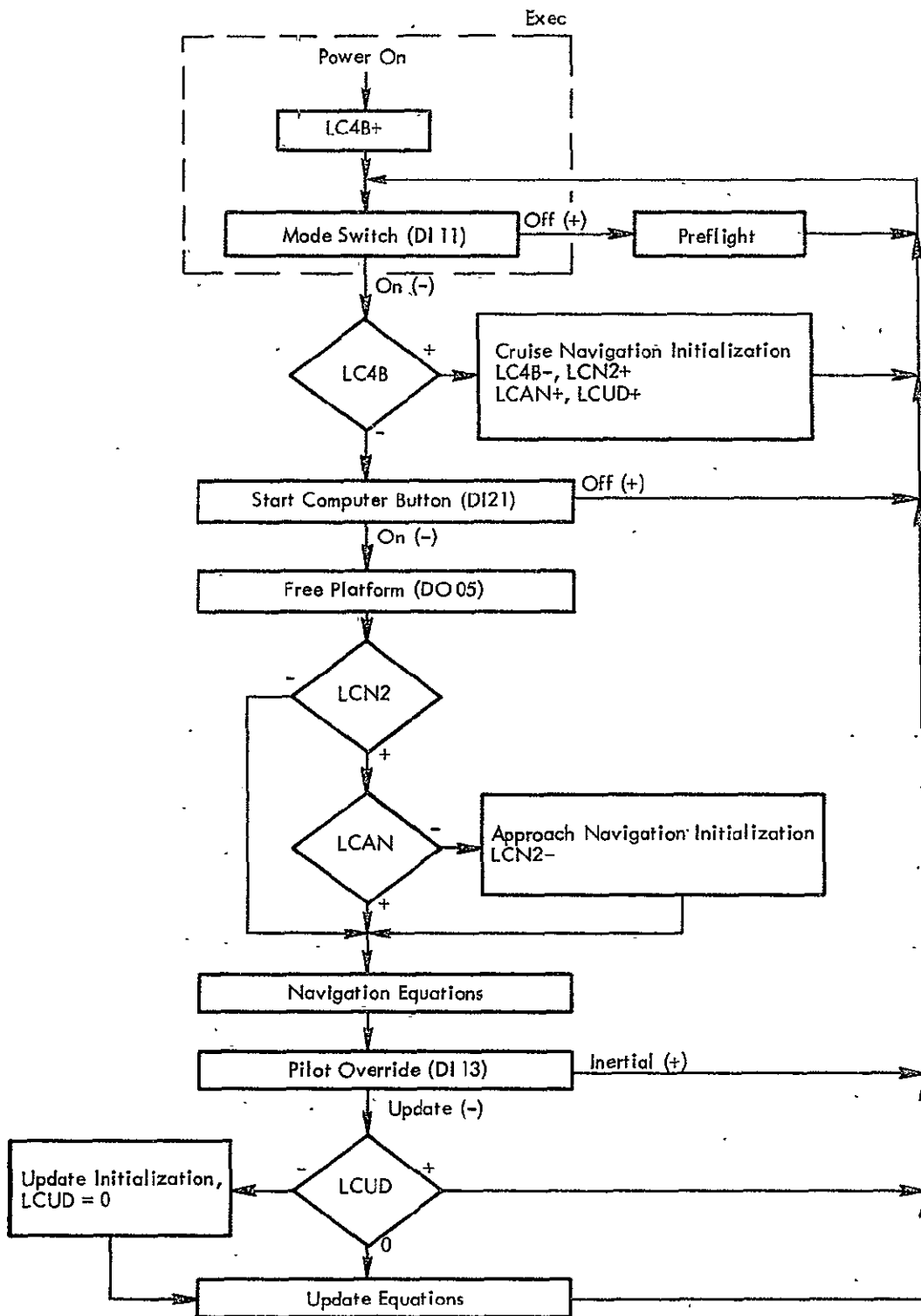


Figure 5-1 Simplified Functional Flow

The radar update option may be switched in and out at the users discretion simply by transmitting the proper LCUD command. Once LCAN is set negative, however, navigation is locked in the approach frame.

Equation/Program Verification

The number and complexity of the equations dictates simulation as the primary verification tool. This is done at two levels, the Fortran level and the operational program level, the latter providing a bit by bit comparison to the actual Gemini computer execution.

The simulation programs are implementated as indicated in Figure 5-2, where only the navigation portion is shown. The lower half of the figure is the actual navigation equation mechanization as derived in Appendix A (Figure A-4); and the upper half simulates the accelerations that the inertial platform experiences. The reference accelerations are defined in the navigation frame and the resulting accelerations in the platform frame are computed. The reference accelerations may be redefined as a function of time to allow simulation of realistic flight profiles. By evaluating the exact integrals of the acceleration forcing function, which is specified in the earth surface fixed frame, reference solutions for position and velocity are obtained. This approach allows accurate evaluation of computational/mechanization errors. For example, one of the options replaces the spherical earth gravity model in the

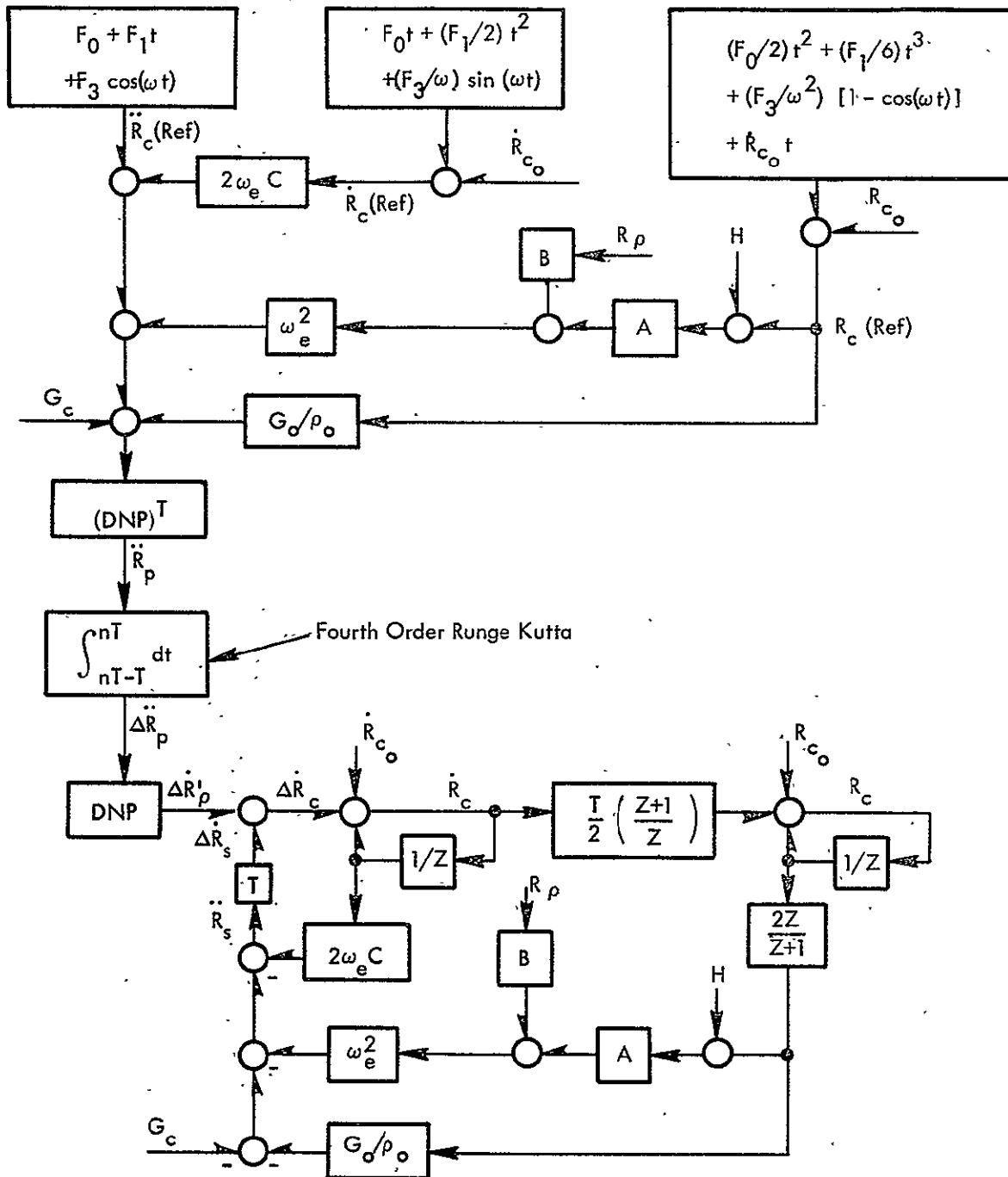


Figure 5-2 System Simulation Implementation

forcing function with a very accurate obtate spheroid model, thus, providing a measure of the accuracy of the navigation model used. The exact integrals of the reference acceleration are also used to simulate the radar updates transmitted to the navigation system.

Typical Fortran simulation results are shown in Figures 5-3 through 5-6. These results shown mechanization and computational errors only, i. e., a perfect IMU. The flight profile is plotted in Figure 5-3. The vehicle accelerates for ten seconds to a vertical velocity (\dot{Z}_c) of ten feet per second and a down range velocity (\dot{X}_c) of 100 feet/second. This constant velocity is held to an altitude of 500 feet where the vertical velocity is reduced to zero. After 200 seconds of flight, navigation is switched to the approach frame.

At 200 seconds the position errors before and after switching to approach are:

	<u>Cruise</u>	<u>Approach</u>
ΔX_c	0.5 feet	-1.1 feet
ΔY_c	-.9 feet	-15.5 feet
ΔZ_c	34.4 feet	14.0 feet

The error build up prior to frame change is totally due to mechanization and computation which is performed in floating point, single precision on the IBM 360-50 computer. The frame change errors result from

5-7

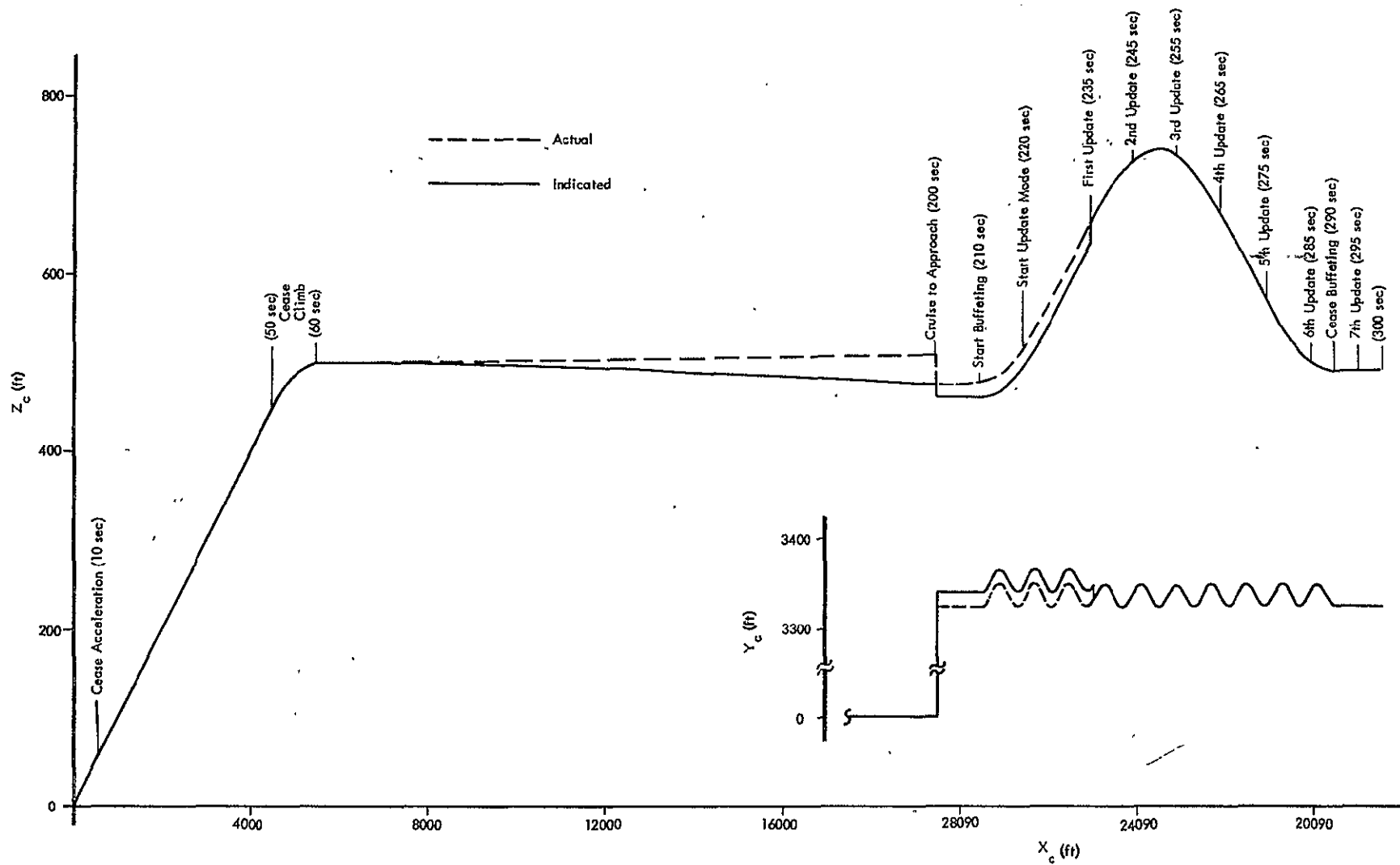


Figure 5-3 Flight Profile

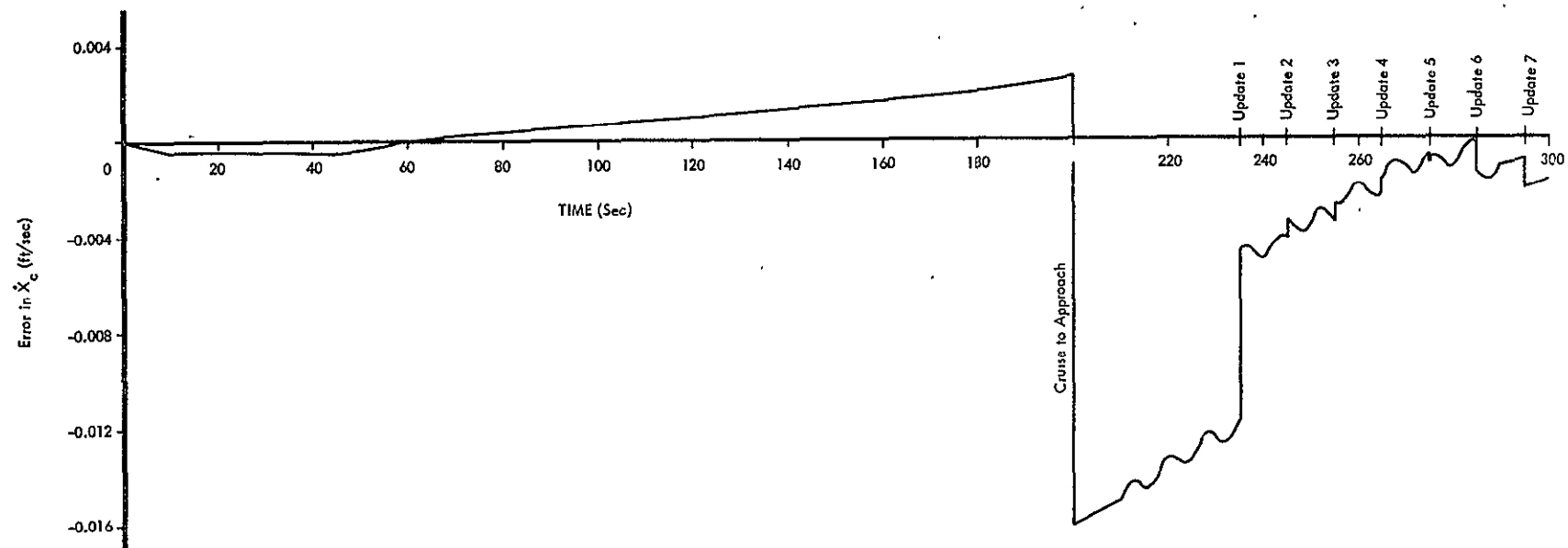


Figure 5-4 X Channel Velocity Error

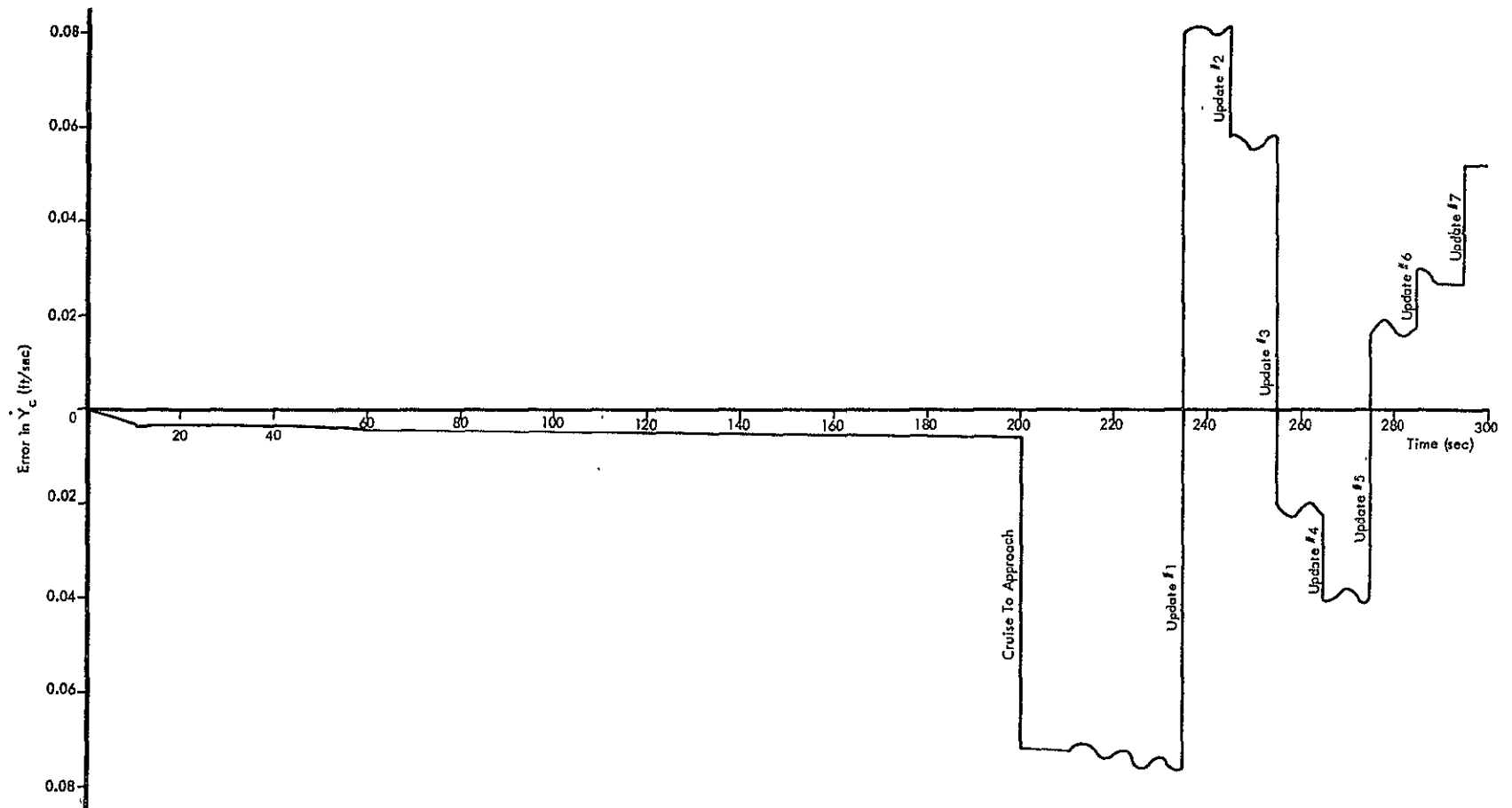


Figure 5-5 Y Channel Velocity Error

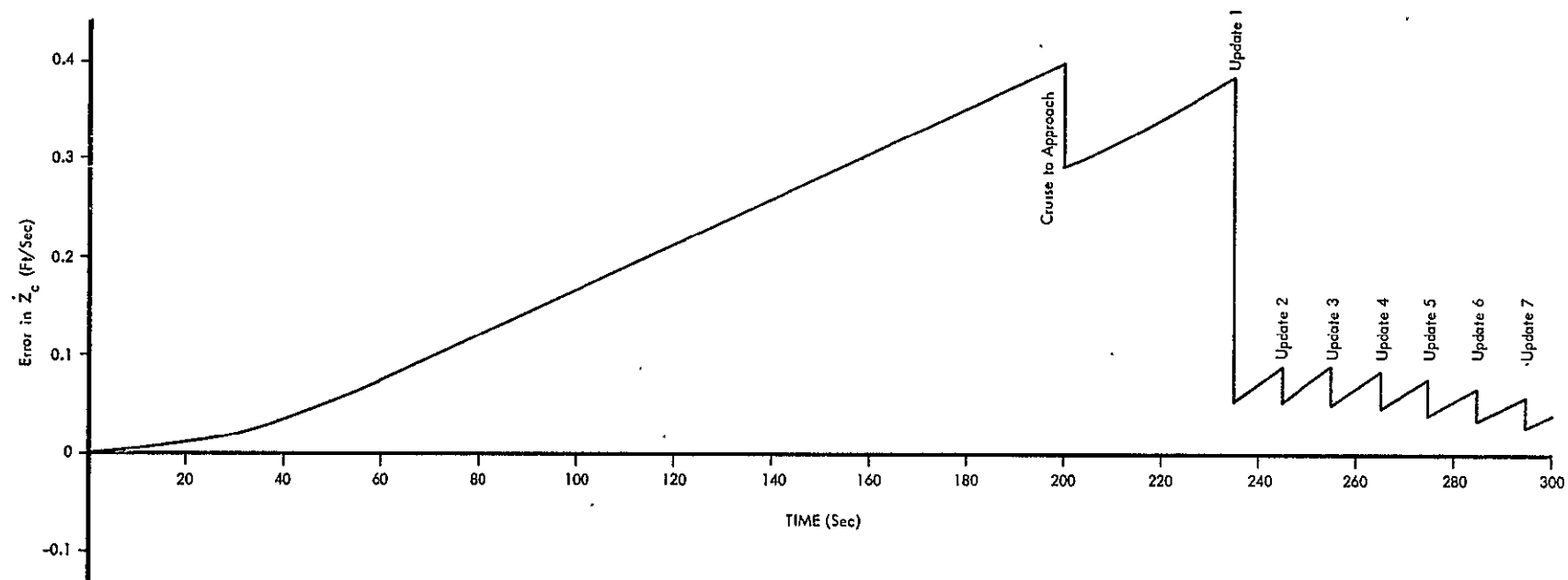


Figure 5-6 Z Channel Velocity Error

truncation due to the large numbers (earth's radii) required for the transformation.

As shown in Figure 5-3, buffeting is introduced at 210 seconds with a slow frequency large amplitude sine wave in the vertical channel, and high frequency smaller amplitude sine wave in the horizontal channel. The first update at 235 seconds removes essentially all the accrued error. The updates continue at ten second intervals with Z_0 being set equal to 0.7.

The plots in Figures 5-4, 5-5, and 5-6 show the velocity error growths and the update corrections. The erratic behavior of the y channel updates is due primarily to the error resulting from the integration algorithm used to compute R_{cc} for the preprocessor. (The integration algorithm assumes a constant acceleration over the one second computation cycle.) Since it does not accurately integrate the high frequency sine wave buffeting, a component of acceleration is allowed to enter the preprocessor filter which is designed to recognize only linear motion. The z channel bias offset of about 0.05 fps is also due primarily to the integration algorithm error.

After completing the Fortran simulation level verification, the Gemini computer instructions are simulated via the operational program simulator. The primary difference between the two should be a slight loss of accuracy due to the Gemini computer's fixed point arithmetic.

A typical comparison of results is shown in Figure 5-7 for a thousand seconds of stationary navigation, where the operational simulation error growth is slightly faster. The dominate error source for this case is the simulated quantization (0.1 feet/second) of the accelerometer outputs. This is illustrated by the short time history of x channel velocity in Figure 5-8.

The final program verification is performed using the Gemini computer itself. The flight software includes a self-test feature, controlled by MDIU insertable logical choices, which exercises essentially all of the program loops without the necessity of tying the computer into the other flight system hardware. The results are verified by comparisons to the previously generated simulation results.

This final program checkout is actually accomplished in three levels; the first two utilizing the self-test feature. These levels are:

Level 1: This test, which checks computer and software only, has completely repeatable results which compare bit by bit to the operational simulation results. This is possible because the self-test program generates its own accelerometer data and fixes the computation cycle to exactly one second.

Level 2: The software "fixed" computation cycle is included in the program execution. Since the software computation cycle

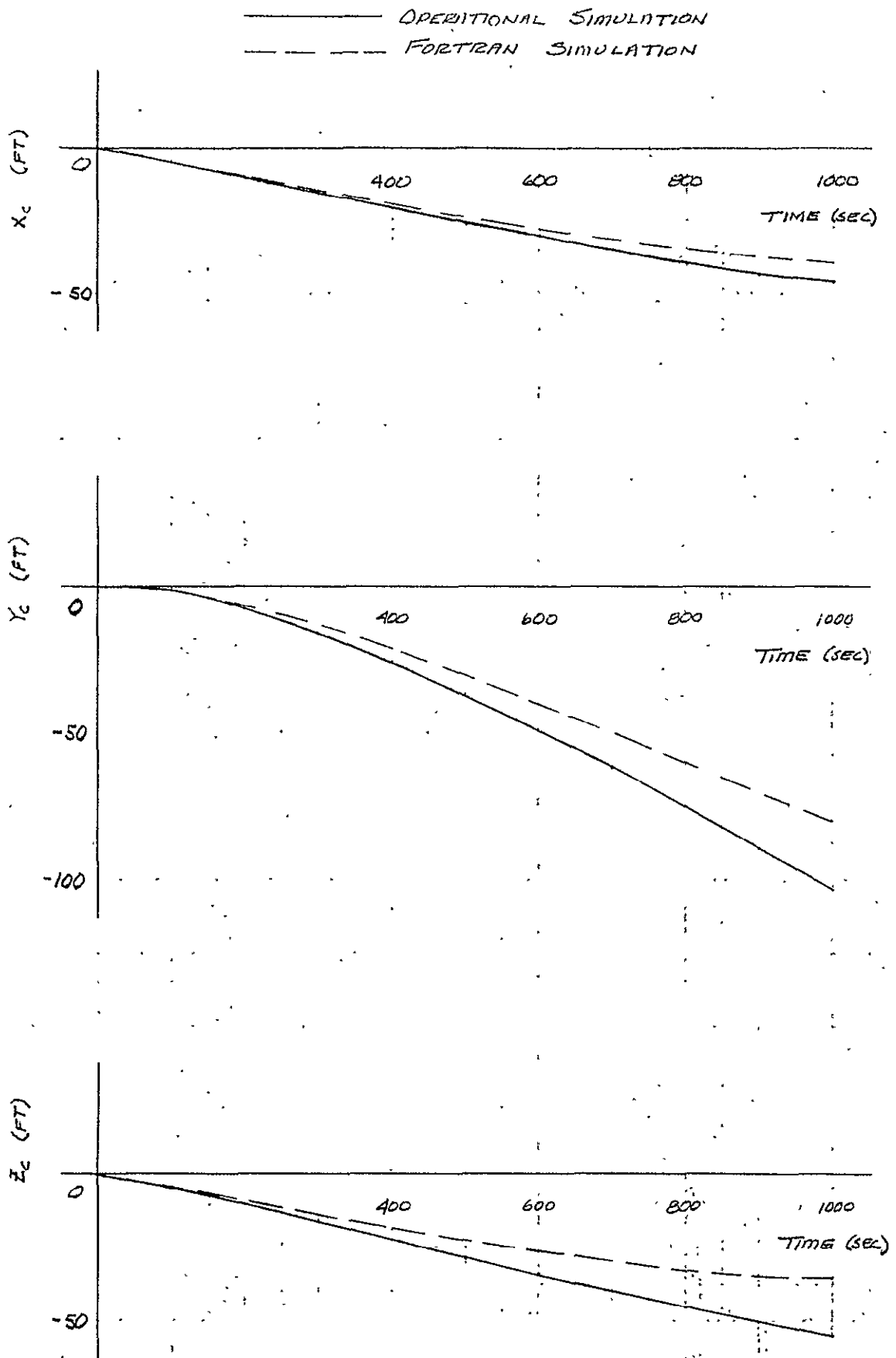


Figure 5-7 Stationary Navigation Simulation Results

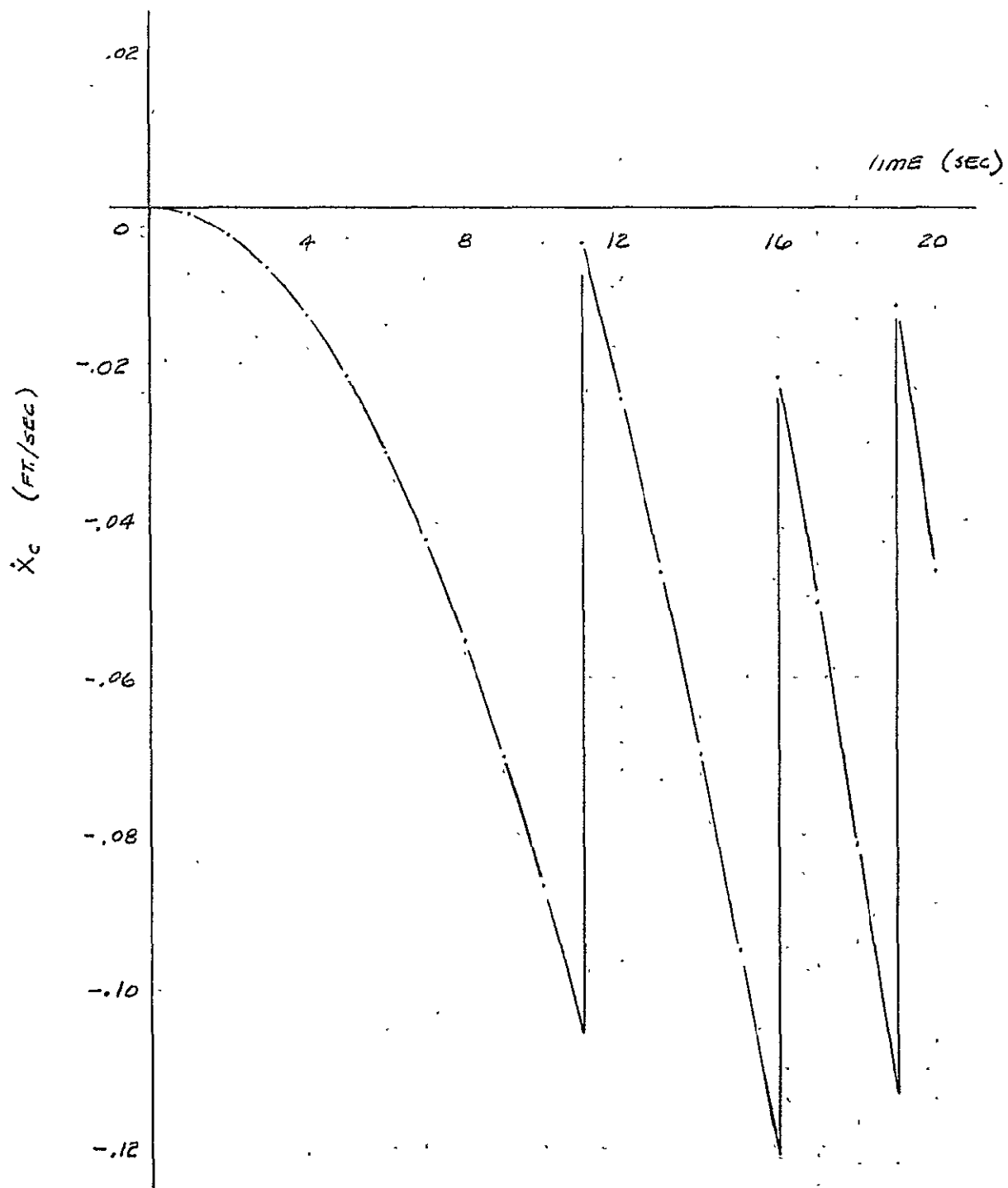


Figure 5-8 Effect of Accelerometer Quantization on Velocity

control results in slightly asynchronous operation, this test assures that the deviations are insignificant.

Level 3: Acceleration data is generated by the AGE equipment which allows testing of the interface as well as the operational software.

The self-test feature has the additional advantage that the program can be rechecked at any time in the field.

Section 6

POST FLIGHT ANALYSIS

Post flight analysis was necessarily limited by the small amount of data supplied; i. e., a partial data set for each of two flights. Its fragmented nature further hindered the analysis because of the limited number of continuous blocks spanning a sufficient time interval, e. g., 200 seconds.

Of primary interest was the accuracy of the velocity update and the radar noise model. An indication of the radar noise may be obtained from the preprocessor ΔR , the three elements of which were telemetered down and recorded at IBM's request.

As defined in Section 4,

$$(6-1) \quad R = R_r - R_p = \text{radar value} - \text{predicted radar value}$$

where

$$(6-2) \quad R_p = \hat{R} + T\hat{R} = \text{previous estimate of position} + \text{velocity estimate} \\ \text{times } T$$

The variance of ΔR is obtained by computing the expectation of ΔR^2

which gives

$$(6-3) \quad \sigma_{\Delta R}^2 = \sigma_{R_r}^2 + \sigma_{\hat{R}}^2 + T^2 \sigma_{\hat{R}}^2 + 2T K_c \sigma_{\hat{R}} \sigma_{\hat{R}}$$

where K_c is the correlation coefficient given in Figure 4-5. For $Z_o = 0.7$, $K_c = .039$, and referring to Figures 4-1 and 4-2 $\sigma_{\hat{R}} \approx 0.62 \sigma_{R_r}$ and $\sigma_{\hat{R}} \approx 0.1 \sigma_{R_r}$. Substituting into equation (6-3) yields

$$(6-4) \quad \sigma_{\Delta R}^2 = 1.40 \sigma_{R_r}^2 \text{ or } \sigma_{R_r} = 0.85 \sigma_{\Delta R}$$

Before analyzing the recorded DAS data, which is transmitted at 2.4 second intervals, linear extrapolation was used to reconstruct a sequence with uniform one second intervals. This was performed on the data supplied for flight number E2-03. The standard deviations and normalized autocorrelations for the preprocessor ΔR were computed and the results are shown in Figures 6-1, 6-2, and 6-3. These are based on 150 seconds of data recorded during the approach update phase. The autocorrelation was computed from equation 6-5.

$$(6-5) \quad \varphi_{\Delta x \Delta x} = \frac{1}{150} \sum_{i=1}^{150} \Delta X_i(t) \Delta X_i(t + \tau)$$

The largest standard deviation ($\sigma_{\Delta x}$) is about 2.5 feet. According to Equation 6-4, the radar noise then has a standard deviation on the order of two feet, which is much smaller than originally anticipated. These results are fairly well verified by the plots of velocity after touchdown shown in Figure 6-4. Since the system is performing stationary navigation, the ideal velocity is zero. The actual velocities, although biased off by the IMU errors, indicates an update accuracy on the order of that

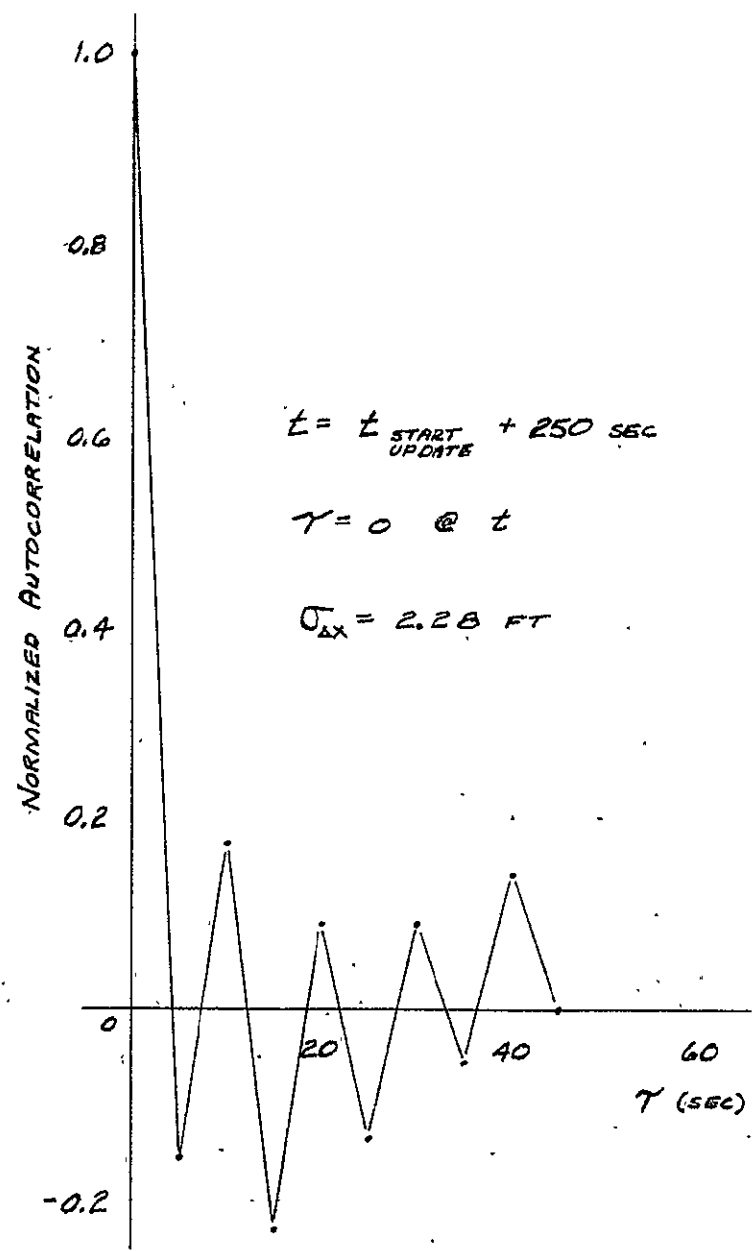
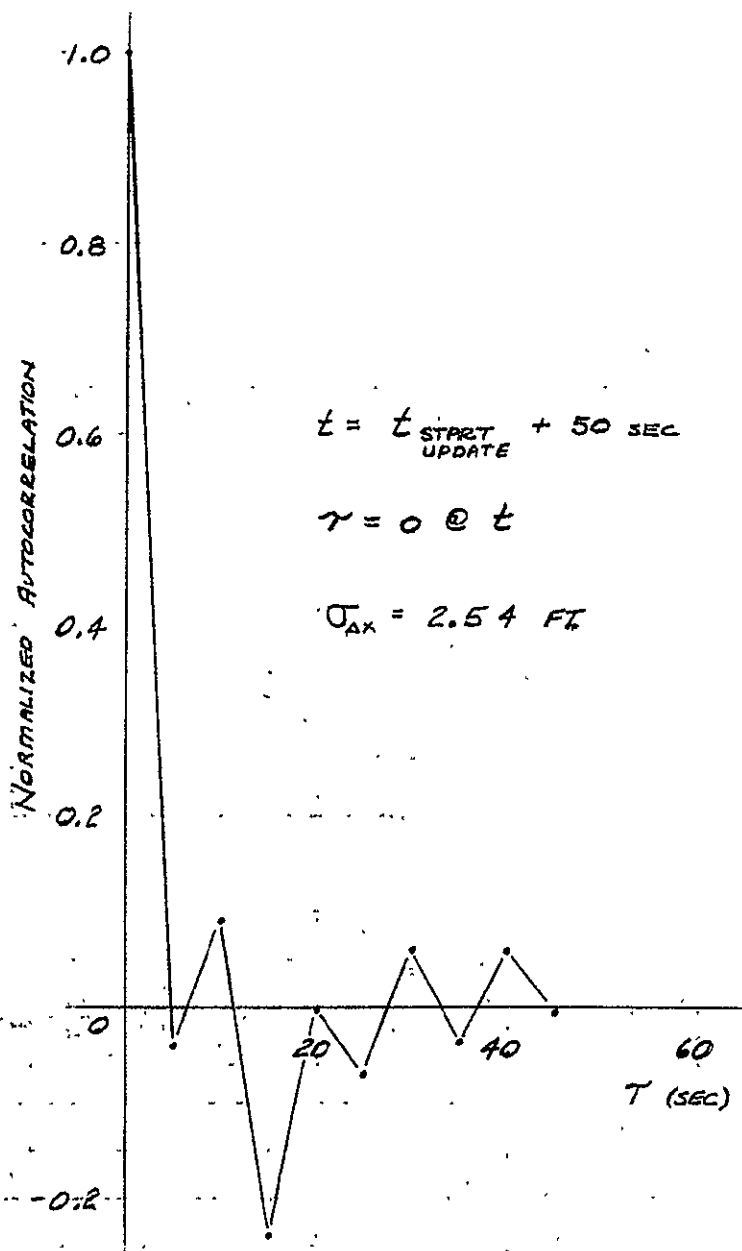
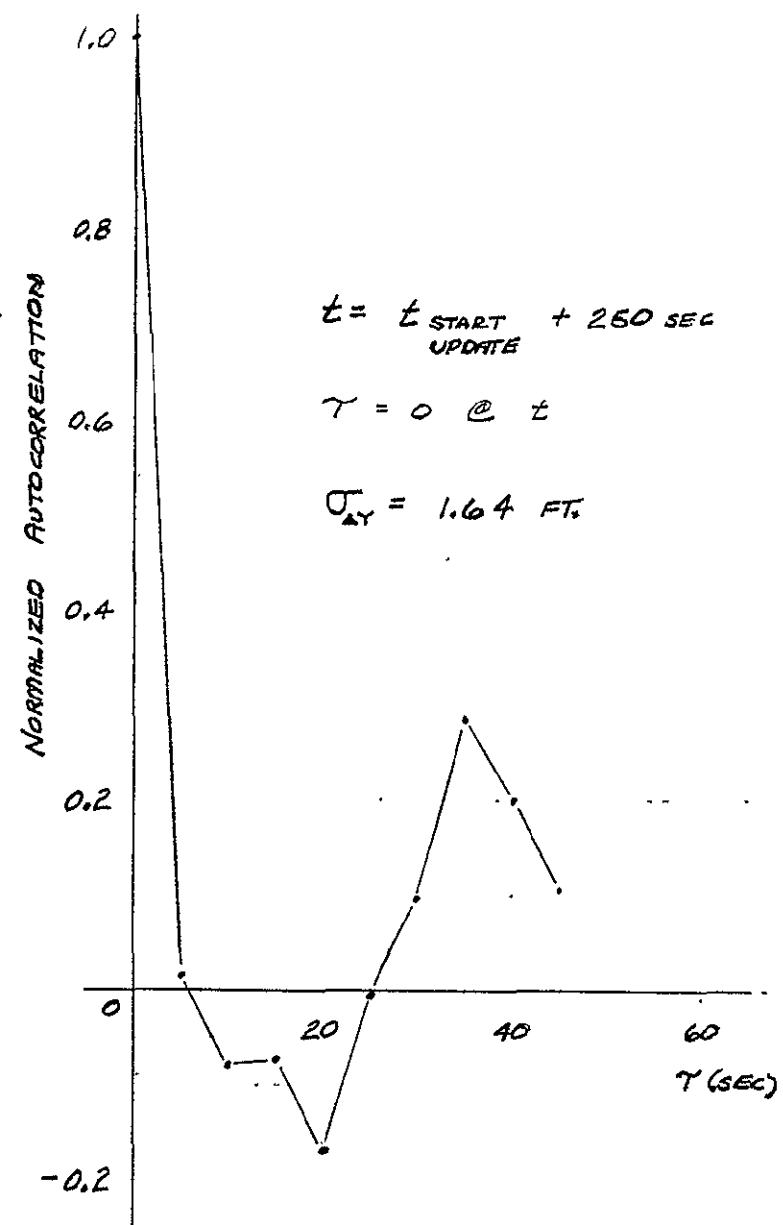
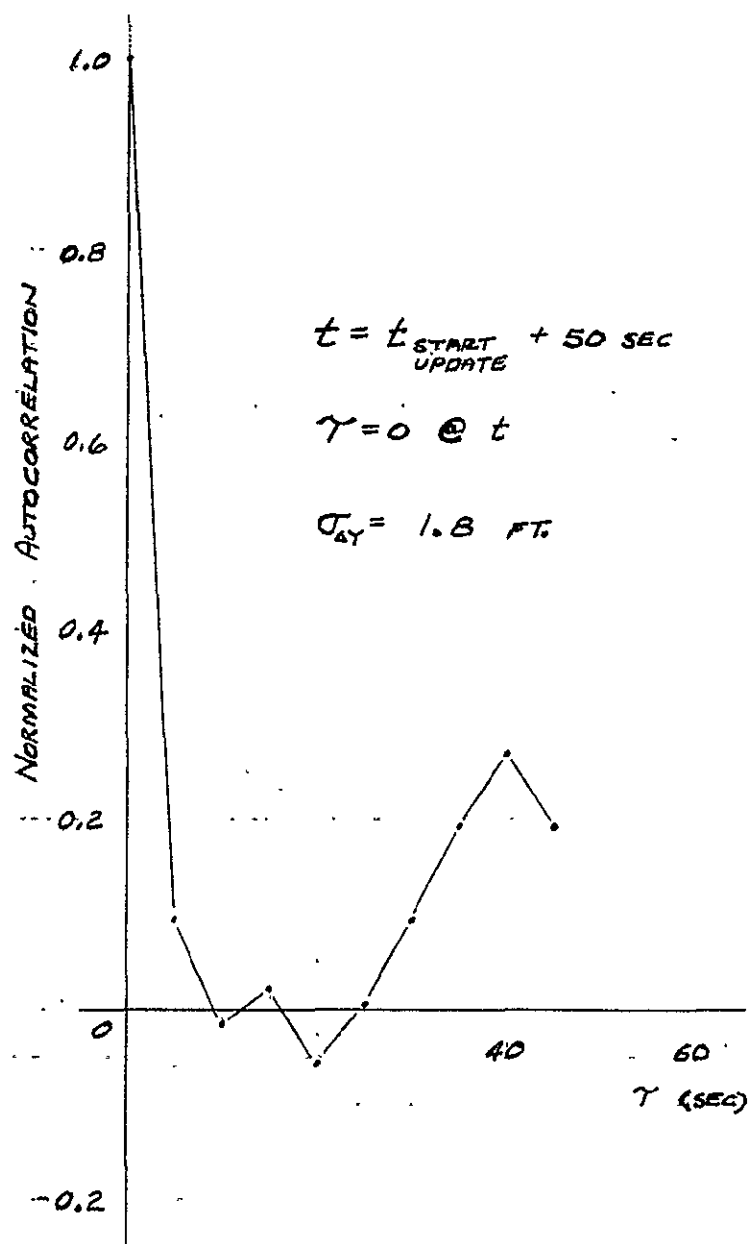
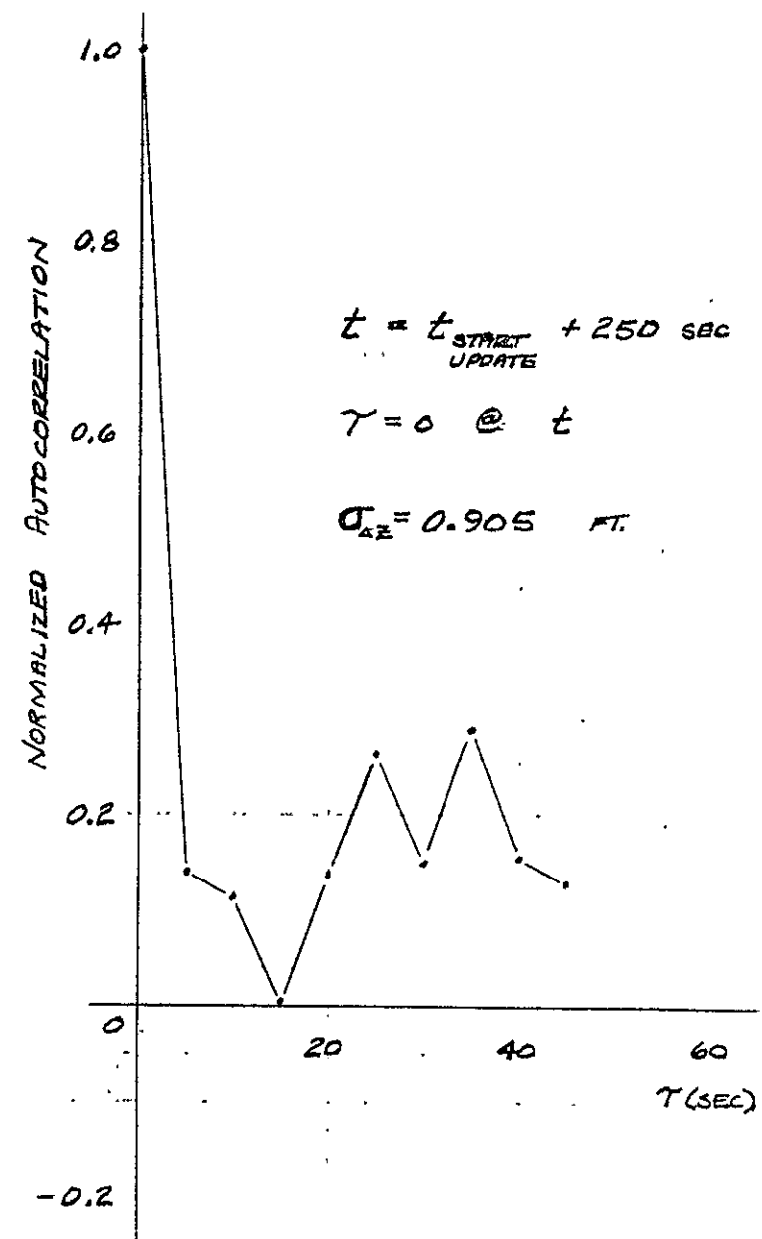
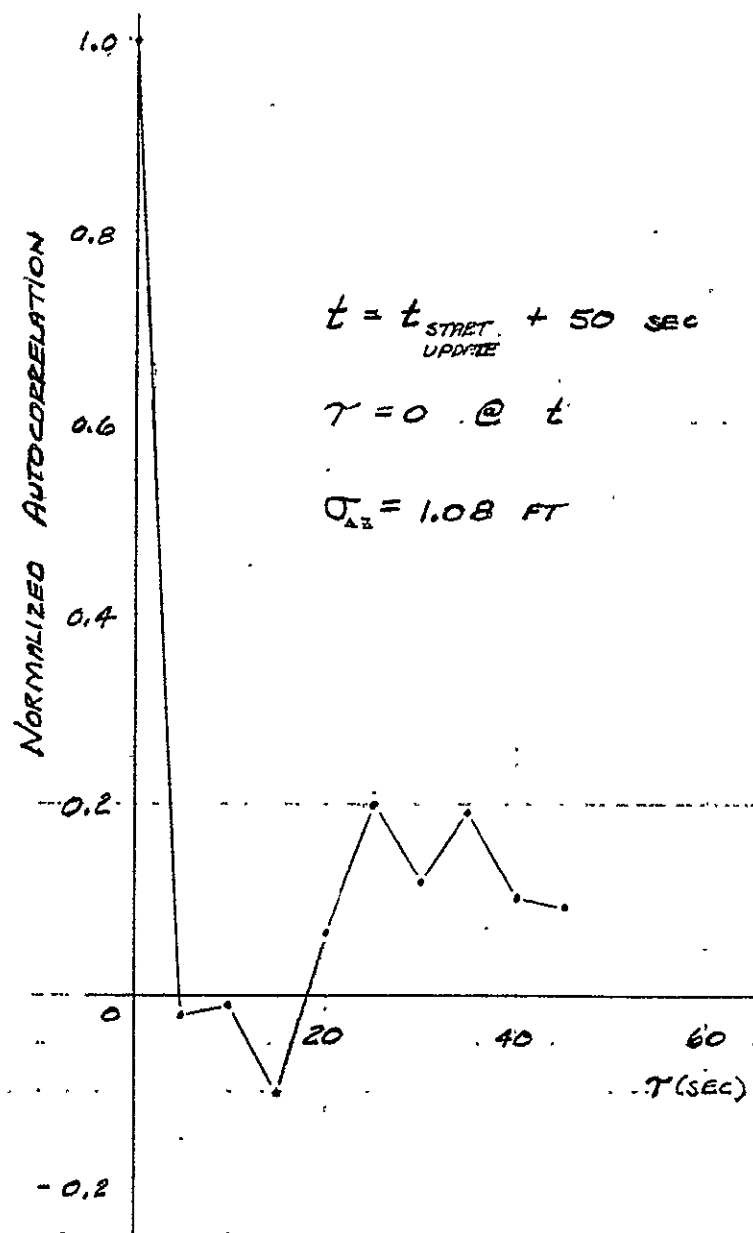


Figure 6-1 Normalized Autocorrelation for Preprocessor ΔX

Figure 6-2 Normalized Autocorrelation for Preprocessor ΔY

Figure 6-3 Normalized Autocorrelation for Preprocessor ΔZ

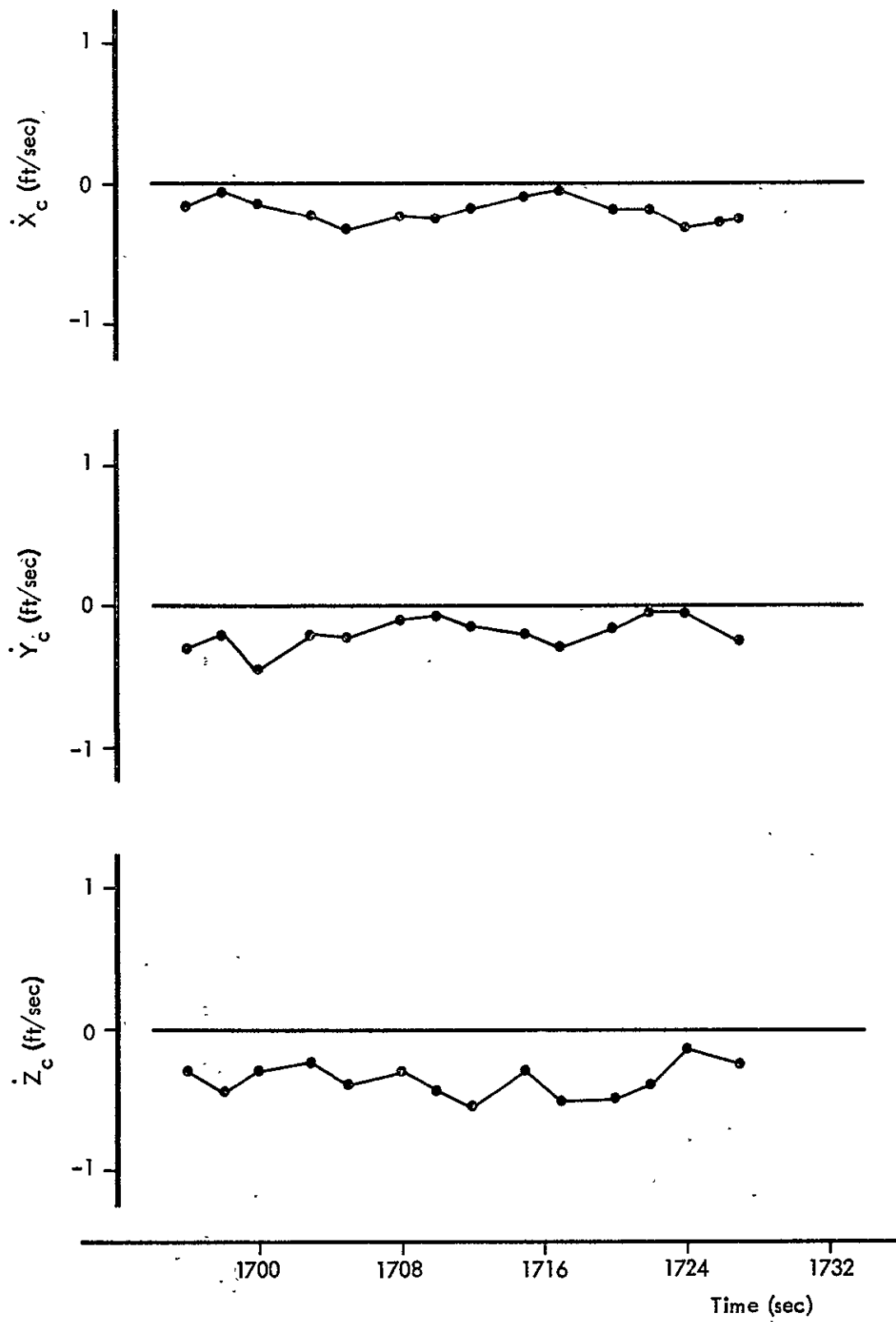


Figure 6-4. Navigation Velocity After Landing

predicted by Figure 4-2 for the above radar noise model, i. e., approximately 0.2 fps.

The plots in Figures 6-5, 6-6, and 6-7 are from data supplied for a case where the system is being updated while performing stationary navigation on the pad after 2000 seconds of flight. The only noise is that due to the IMU and occasional quantization jumps in the radar data. It thus provides a good demonstration of the preprocessor operation.

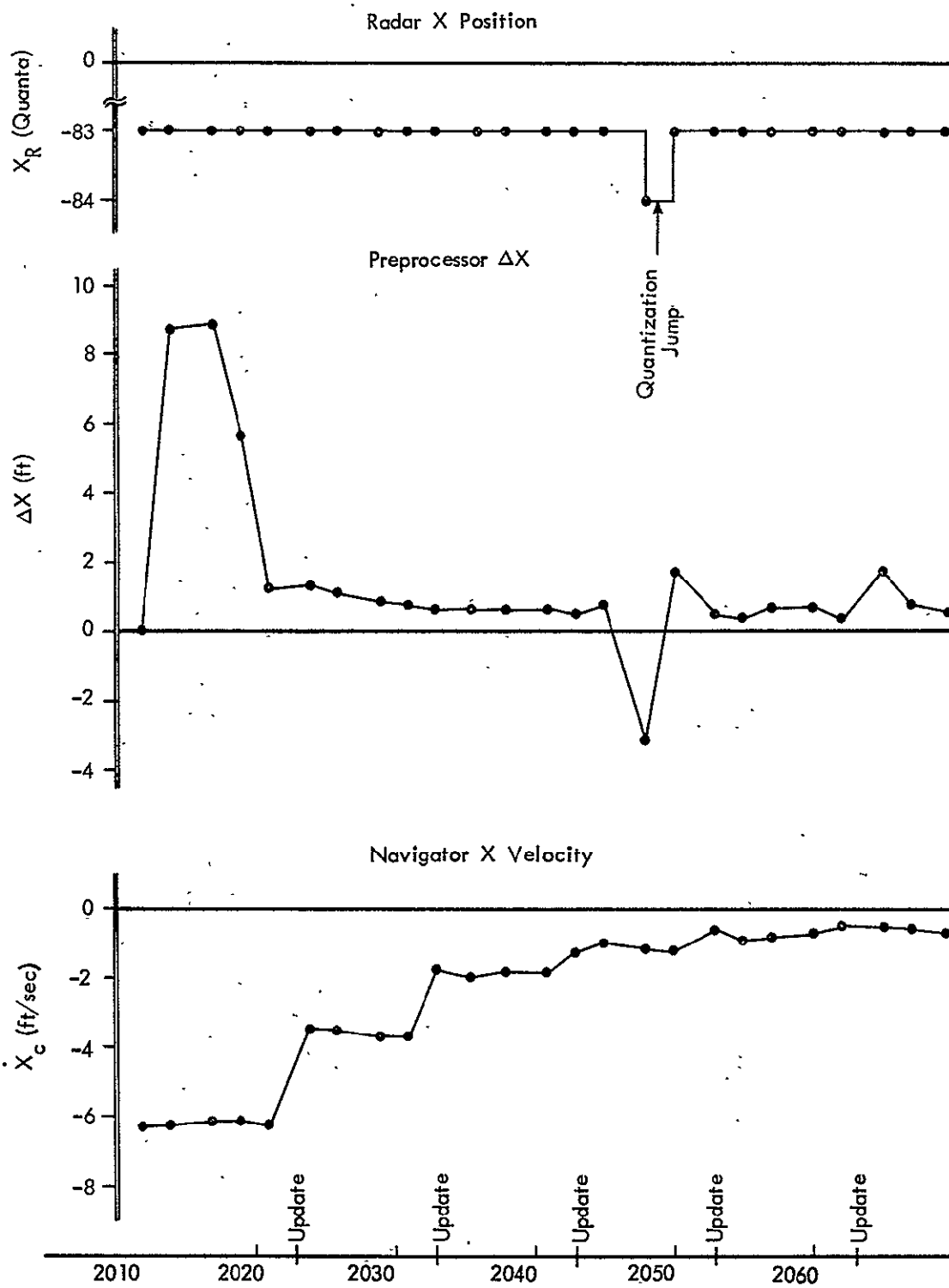


Figure 6-5. X Channel Stationary Navigation Quantities

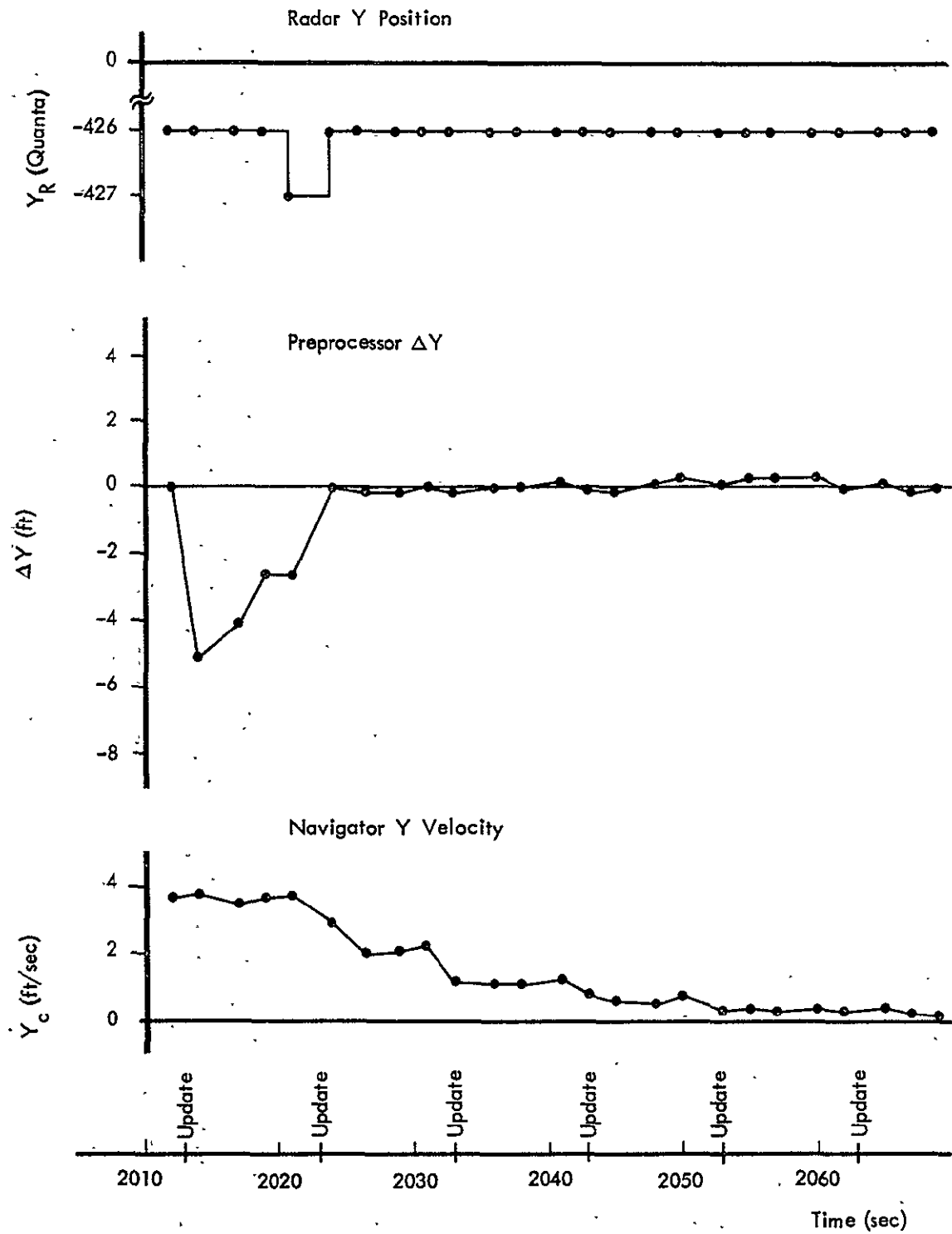


Figure 6-6. Y Channel Stationary Navigation Quantities

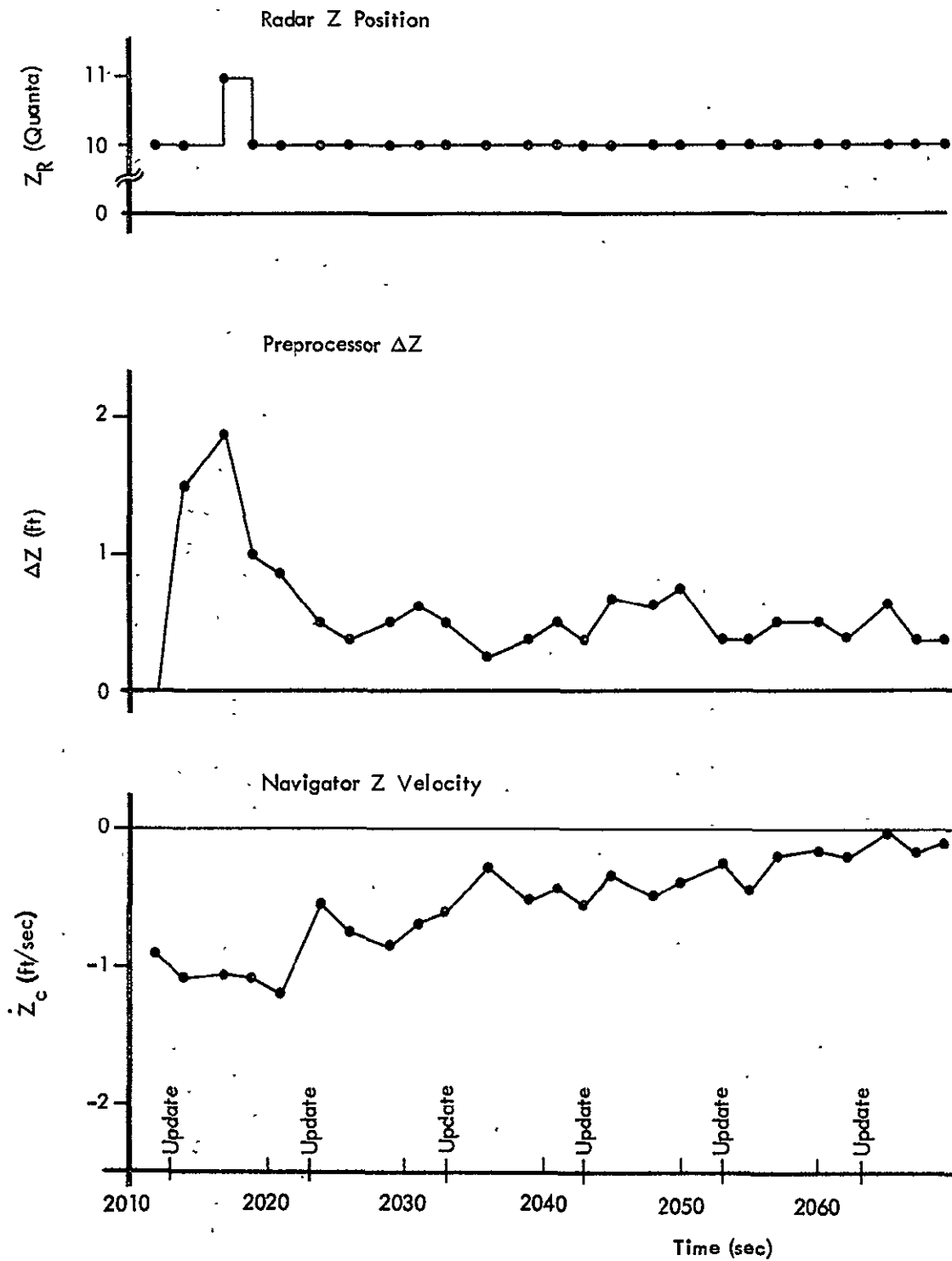


Figure 6-7. Z Channel Stationary Navigation Quantities

Appendix A

NAVIGATION EQUATIONS DERIVATION

As described in Section 3, navigation is performed in either of two earth-surface-fixed rectangular coordinate frames; the cruise frame with its origin at the liftoff point, or the approach frame with its origin at the touchdown point. The inertial platform and cruise navigation axes are indicated in Figure A-1. At liftoff ($t = 0$) the origins of the platform frame (X_p, Y_p, Z_p) and the navigation frame (X_c, Y_c, Z_c) are coincident. The platform axes are nominally aligned North, East and down (plumb bob vertical); however, as shown in Figure A-1, any arbitrary azimuth angle (ϕ_p) is equally acceptable. The navigation frames are nominally aligned North, East, and up (plumb bob vertical), and either or both may also be aligned at arbitrary azimuth angles with respect to the North-East nominal. A positive azimuth is always defined as a positive rotation about the axis along the plumb bob vertical of the frame.

With proper initialization, the same equations apply equally to either navigation frame. Therefore, no distinction, as to cruise or approach, is necessary in the ensuing derivation. Referring to Figure A-1, the following reference frames may be defined.

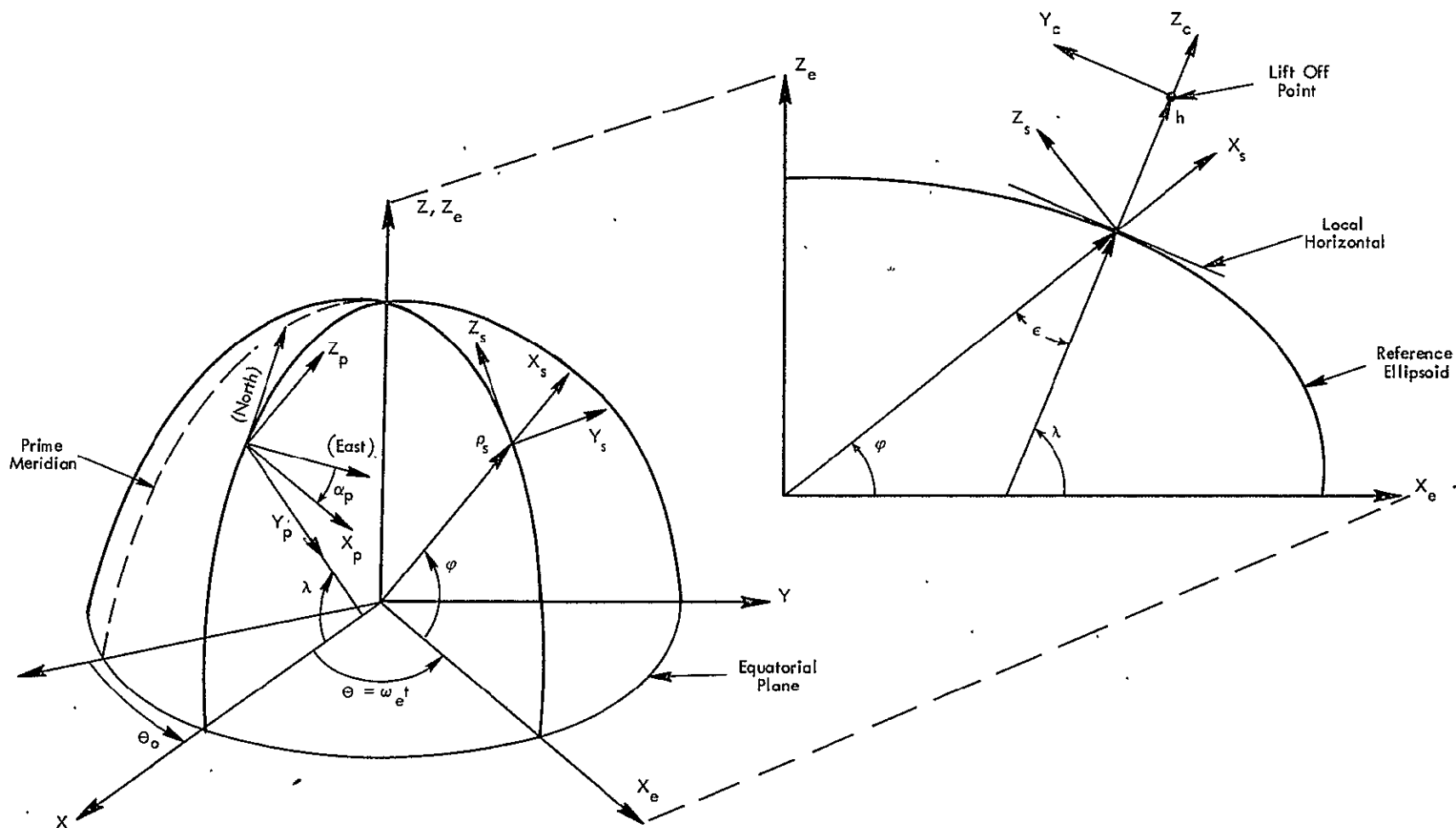


Figure A-1 Coordinate Axes

$$(A-1) \quad R = \begin{vmatrix} X \\ Y \\ Z \end{vmatrix} \quad \text{Earth Centered Inertial}$$

$$(A-2) \quad R_e = \begin{vmatrix} X_e \\ Y_e \\ Z_e \end{vmatrix} \quad \text{Earth Centered Fixed}$$

$$(A-3) \quad R_s = \begin{vmatrix} X_s \\ Y_s \\ Z_s \end{vmatrix} \quad \begin{array}{l} \text{Geocentric Earth (Reference Spheroid)} \\ \text{Surface Fixed} \end{array}$$

$$(A-4) \quad R_c = \begin{vmatrix} X_c \\ Y_c \\ Z_c \end{vmatrix} \quad \text{Geodetic Earth Surface Fixed}$$

Also define

$$(A-5) \quad H = \begin{vmatrix} 0 \\ 0 \\ h \end{vmatrix} \quad \begin{array}{l} \text{Altitude of } R_c \text{ Frame Origin Above} \\ \text{Reference Spheroid} \end{array}$$

and

$$(A-6) \quad R_\rho = \rho_s \begin{vmatrix} \cos \phi \\ 0 \\ \sin \phi \end{vmatrix}$$

The following matrix relationships may now be defined.

$$(A-7) \quad R_e = N R$$

where $N = f(\Theta)$

$$(A-8) \quad R_s = M (R_e - R_\rho)$$

where $M = f(\phi) = \text{constant matrix}$

$$(A-9) \quad R_c = L R_s - H$$

where $L = f(\epsilon, \alpha) = \text{constant matrix}$

Combining Equations (A-7), (A-8), and (A-9)

$$(A-10) \quad R_c = -H - L M R_\rho + L M N R$$

Differentiating Equation (A-10) twice

$$(A-11) \quad \dot{R}_c = L M (N \dot{R} + \dot{N} R)$$

$$(A-12) \quad \ddot{R}_c = L M (N \ddot{R} + 2\dot{N} \dot{R} + \ddot{N} R)$$

From Equation (A-10)

$$(A-13) \quad R = N^T (R_\rho + M^T L^T (R_c + H))$$

and from Equations (A-11) and (A-13)

$$(A-14) \quad \dot{R} = N^T M^T L^T \dot{R}_c - N^T \dot{N} N^T (R_\rho + M^T L^T (R_c + H))$$

Substituting (A-13) and (A-14) into (A-12)

$$(A-15) \quad \ddot{R}_c = L M (\ddot{N} N^T - 2(\dot{N} N^T)^2 (R_\rho + M^T L^T (R_c + H)) \\ + 2\dot{N} N^T M^T L^T \dot{R}_c + N \ddot{R})$$

Equation (A-15) may be simplified as follows

$$(A-16) \quad N = \begin{vmatrix} \cos \Theta & \sin \Theta & 0 \\ -\sin \Theta & \cos \Theta & 0 \\ 0 & 0 & 1 \end{vmatrix} \quad \text{Where } \Theta = \omega_e t$$

Differentiating N and defining the ∇ operator

$$(A-17) \quad \dot{N} = -\omega_e \begin{vmatrix} \sin \Theta & -\cos \Theta & 0 \\ \cos \Theta & \sin \Theta & 0 \\ 0 & 0 & 0 \end{vmatrix} = -\omega_e \begin{vmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix} N \equiv -\omega_e \nabla N$$

Differentiating again

$$(A-18) \quad \ddot{N} = -\omega_e \nabla \dot{N} = \omega_e^2 \nabla^2 N$$

Substituting (A-17) and (A-18) into (A-15) yields

$$(A-19) \quad \ddot{R}_c = L M (-\omega_e^2 \nabla^2 (R_p + M^T L^T (R_c + H)) \\ - 2\omega_e \nabla M^T L^T \dot{R}_c + N \ddot{R})$$

Further simplifications are possible from the following identities.

$$(A-20) \quad \nabla^2 = \begin{vmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{vmatrix} = \begin{vmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{vmatrix} + \begin{vmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{vmatrix} \\ = -I + \delta$$

Therefore,

$$(A-21) \quad L M \nabla^2 (L M)^T = -I + L M \delta (L M)^T$$

and defining

$$(A-22) \quad LM = \begin{vmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{vmatrix}$$

then

$$(A-23) \quad LM \delta (LM)^T = \begin{vmatrix} d_{13}^2 & d_{23} d_{13} & d_{33} d_{13} \\ d_{13} d_{23} & d_{23}^2 & d_{33} d_{23} \\ d_{13} d_{33} & d_{23} d_{33} & d_{33}^2 \end{vmatrix}$$

and

$$(A-24) \quad LM \nabla (LM)^T = \begin{vmatrix} 0 & d_{12} d_{21} - d_{11} d_{22} & d_{12} d_{31} - d_{11} d_{32} \\ d_{11} d_{22} - d_{12} d_{21} & 0 & d_{31} d_{22} - d_{21} d_{32} \\ d_{11} d_{32} - d_{12} d_{31} & d_{21} d_{32} - d_{31} d_{22} & 0 \end{vmatrix}$$

With the following definitions

$$(A-25) \quad D = LM$$

$$(A-26) \quad A = -I + D \delta D^T$$

$$(A-27) \quad B = D \nabla^2$$

$$(A-28) \quad C = D \nabla D^T$$

Equation (A-19) may be rewritten as

$$(A-29) \quad \ddot{\mathbf{R}}_c = -\omega_e^2 (B \mathbf{R}_\rho + A (\mathbf{R}_c + H)) - 2\omega_e C \dot{\mathbf{R}}_c + D N \ddot{\mathbf{R}}$$

The elements of the matrix N have been previously defined. It can be readily shown that the elements of the remaining matrices in Equation (A-29) are the following.

$$(A-30) \quad A = \begin{vmatrix} \cos^2 \lambda & \sin^2 \alpha & -1 & \cos^2 \lambda & \cos \alpha & \sin \alpha & \cos \lambda & \sin \lambda & \sin \alpha \\ \cos^2 \lambda & \cos \alpha & \sin \alpha & \cos^2 \lambda & \cos^2 \alpha & -1 & \cos \lambda & \sin \lambda & \cos \alpha \\ \cos \lambda & \sin \alpha & \sin \alpha & \cos \lambda & \sin \lambda & \cos \alpha & -\cos^2 \lambda \end{vmatrix}$$

$$(A-31) \quad B = \begin{vmatrix} \sin \lambda & \sin \alpha & -\cos \alpha & 0 \\ \sin \lambda & \cos \alpha & \sin \alpha & 0 \\ -\cos \lambda & & 0 & 0 \end{vmatrix}$$

$$(A-32) \quad C = \begin{vmatrix} 0 & -\sin \lambda & \cos \lambda & \cos \alpha \\ \sin \lambda & 0 & -\cos \lambda & \sin \alpha \\ -\cos \lambda & \cos \alpha & \cos \lambda & \sin \alpha & 0 \end{vmatrix}$$

$$(A-33) \quad D = \begin{vmatrix} -\sin \lambda & \sin \alpha & \cos \alpha & \cos \lambda & \sin \alpha \\ -\sin \lambda & \cos \alpha & -\sin \alpha & \cos \lambda & \cos \alpha \\ \cos \lambda & & 0 & \sin \lambda \end{vmatrix}$$

The first and second terms of Equation (A-29) are the familiar expressions for centripetal and coriolis accelerations respectively. The remaining term, the system forcing function, is composed of the measured and gravitational accelerations. In the ECI frame this may be expressed as

$$(A-34) \quad DN \ddot{R} = DN (\ddot{R}_m + \ddot{R}_g)$$

The gravity computation for this application can be accomplished with sufficient accuracy by choosing an "equivalent" spherical earth model; the computational frame for which is indicated in Figure A-2. Since the gravity computation is now independent of longitude, Equation (A-34) may be rewritten as

$$(A-35) \quad D N \ddot{R} = D N \ddot{R}_m + D \ddot{R}_g$$

where

$$(A-36) \quad D \ddot{R}_g = D \left(\frac{-GM}{|R_g|^3} \right) R_g$$

and

$$(A-37) \quad R_g = D' \rho_o + D^T R_c$$

where $D' = f(\lambda_g) = \text{constant matrix}$

Equation (A-36) exhibits the well known vertical channel instability. For this application a more desirable solution, which is bounded, can be obtained by setting

$$(A-38) \quad \frac{-GM}{|R_g|^3} = \frac{-G_o}{\rho_o} = \text{constant}$$

then

$$(A-39) \quad D \ddot{R}_g = -G_o D D' - \frac{G_o}{\rho_o} R_c \equiv -G_c - \frac{G_o}{\rho_o} R_c$$

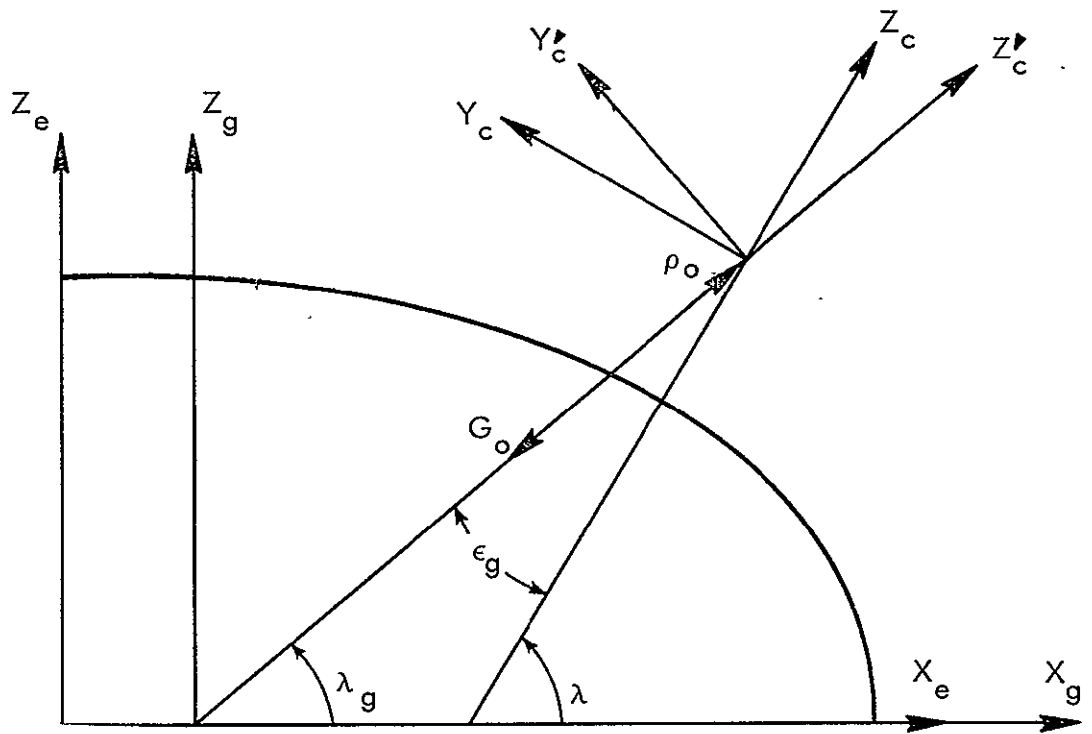


Figure A-2 Gravity Computation Frame

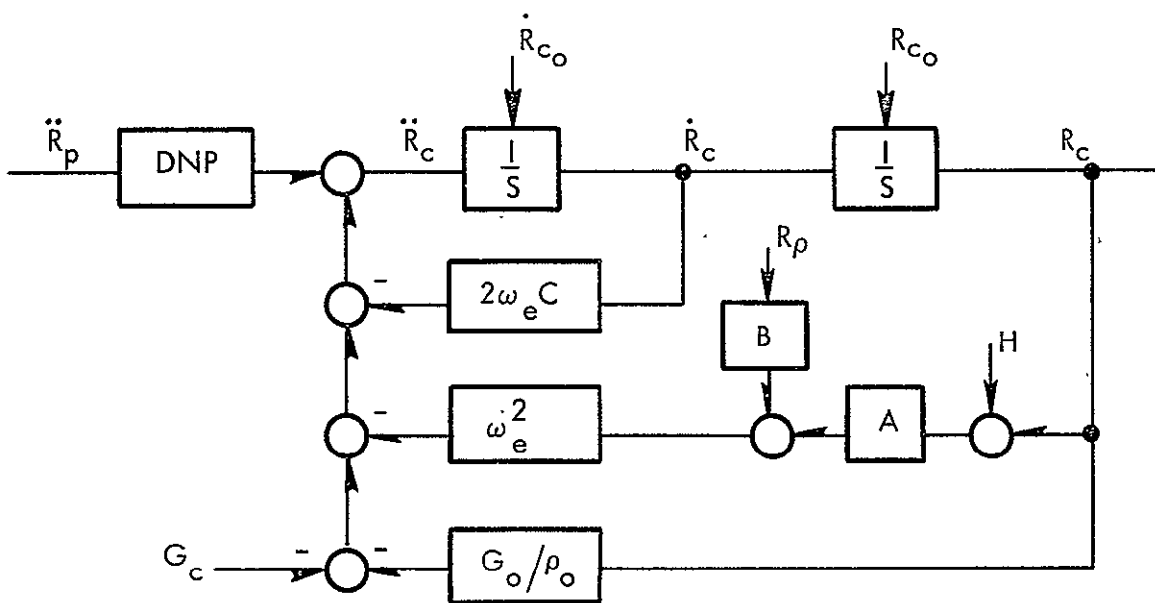


Figure A-3 Ideal Mechanization

It is easily shown that

$$(A-40) \quad G_c = G_o \begin{vmatrix} -\sin \alpha & \sin \epsilon_g \\ -\cos \alpha & \sin \epsilon_g \\ \cos \epsilon_g \end{vmatrix}$$

As indicated in Figure A-1, the inertial platform is aligned to the local vertical at some arbitrary azimuth. Therefore,

$$(A-41) \quad \ddot{R}_m = P \ddot{R}_p$$

where

$$(A-42) \quad P = \begin{vmatrix} \sin \lambda & \sin \alpha_p & -\cos \lambda & -\sin \lambda & \cos \alpha_p \\ \cos \alpha_p & 0 & \sin \alpha_p \\ -\cos \lambda & \sin \alpha_p & -\sin \lambda & \cos \lambda & \cos \alpha_p \end{vmatrix}$$

and

$$(A-43) \quad \ddot{R}_p = \begin{vmatrix} \ddot{X}_p \\ \ddot{Y}_p \\ \ddot{Z}_p \end{vmatrix} \quad (\text{Ideal accelerometer outputs})$$

Introducing Equations (A-35), (A-39), and (A-41) into Equation (A-29), the final equation becomes

$$(A-44) \quad \ddot{R}_c = -\omega_e^2 (B R_p + A (R_c + H)) - 2\omega_e C \dot{R}_c - G_c - \frac{G_o}{\rho_o} R_c + D N P \ddot{R}_p$$

This is shown in block diagram form in Figure A-3.

The equation mechanization shown in Figure A-3 is obviously an idealized version. The actual accelerometer outputs are increments in velocity, and the digital implementation introduces a transport lag in the feedback loops. To offset the error due to this feedback delay, a predictor corrector type integration algorithm is used. The algorithm is derived by assuming a constant acceleration over the integration interval. That is

$$(A-45) \quad \ddot{R}_c = \text{constant}, \quad ((nT - T) < t \leq nT)$$

$$(A-46) \quad \dot{R}_c(nT) = \int_{nT-T}^{nT} \ddot{R}_c dt + \dot{R}_c(nT - T) = \ddot{R}_c T + \dot{R}_c(nT - T)$$

$$(A-47) \quad R_c(nT) = \int_{nT-T}^{nT} (\ddot{R}_c t + \dot{R}_c(nT - T)) dt + R_c(nT - T) \\ = \ddot{R}_c (T^2/2) + \dot{R}_c(nT - T) T + R_c(nT - T)$$

However,

$$(A-48) \quad \ddot{R}_c T \equiv \Delta \dot{R}_c \equiv \Delta \dot{R}'_p + \ddot{R}_s T$$

where

$\Delta \dot{R}'_p$ = accelerometer outputs in the navigation frame

and

\ddot{R}_s = summation of computed accelerations due to gravity, centripetal, and coriolis

Since \ddot{R}_s is treated as a constant over the interval, a reasonable approach is to compute its value at the midpoint of the interval. For the computation

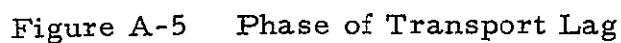
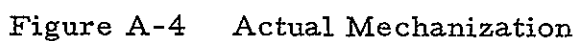
of centripetal and gravitational accelerations, this requires predicting the value R_c forward one-half computation cycle. The most straight forward method is

$$(A-49) \quad R_{cp} = R_c (nT - T) + \dot{R}_c (nT - T) \frac{T}{2}$$

The error, due to the transport lag, in the computation of coriolis acceleration is of sufficiently small magnitude that it is ignored.

The block diagram representation may now be redrawn as shown in Figure A-4. It is of interest to note that the integration algorithm previously derived is the familiar form of trapezoidal integration; which in Z transform notation is $(T/2) \left(\frac{Z+1}{Z-1} \right)$. Trapezoidal integration has perfect phase of -90 degrees (the same as an ideal integrator). Therefore, without the predictor corrector, the solution of the closed loop system is unstable (closed loop poles outside the unit circle) due to the phase lag introduced by the feedback delay $(1/Z)$. This phase lag as a function of frequency is shown in Figure A-5.

The predictor corrector modifies the overall position loop such that it is equivalent, from the standpoint of phase, to replacing the trapezoidal integrators with forward rectangular; yet the more accurate trapezoidal algorithm is retained in the forward loop. The advantage is that forward rectangular integration, which in Z transform notation is $ZT/(Z-1)$, introduces phase lead into the system; and as can be seen from the phase



A-13

characteristic shown in Figure A-6, two forward rectangular integrators provide exactly the phase lead required to offset the phase lag of the feedback delay.

One further correction is required. With an ideal mechanization, the accelerometer outputs in the rotating navigation frame are

$$(A-50) \quad \Delta \dot{R}'_{PI} = \int_{nT-T}^{nT} D N P \ddot{R}_P dt$$

But the physically realizable implementation gives

$$(A-51) \quad \Delta \dot{R}'_P = DNP \int_{nT-T}^{nT} \ddot{R}_P dt$$

where

$$(A-52) \quad \int_{nT-T}^{nT} \ddot{R}_P dt = \Delta \dot{R}_P = \text{accelerometer outputs}$$

Thus the error is

$$(A-53) \quad \epsilon = \Delta \dot{R}'_{PI} - \Delta \dot{R}'_P = D \left(\int_{nT-t}^{nT} NP \ddot{R}_P dt - NP \int_{nT-T}^{nT} \ddot{R}_P dt \right)$$

where

$$N = \begin{vmatrix} \cos(\omega_e t) & \sin(\omega_e t) & 0 \\ -\sin(\omega_e t) & \cos(\omega_e t) & 0 \\ 0 & 0 & 1 \end{vmatrix} \approx \begin{vmatrix} 1 & \omega_e t & 0 \\ -\omega_e t & 1 & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

$$= I + \omega_e t \nabla$$

where

$$\nabla = \begin{vmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{vmatrix}$$

Thus

$$(A-54) \quad NP \simeq P + \omega_e t \nabla P$$

and

$$(A-55) \quad \epsilon = -\omega_e D \nabla P \iint_{nT-T}^{nT} \ddot{R}_P dt dt$$

The corrected navigation frame values are then

$$(A-56) \quad R_P = D \left((P + \omega_e t \nabla P) \int_{nT-T}^{nT} \ddot{R}_P dt + \omega_e \nabla P \iint_{nT-T}^{nT} \ddot{R}_P dt dt \right)$$

Treating \ddot{R}_P as a constant over the interval

$$(A-57) \quad \Delta R_P' = D \left(I + \omega_e \left(t - \frac{T}{2} \right) \nabla \right) P \Delta \dot{R}_P$$

But $(I + \omega_e (t - \frac{T}{2}) \nabla)$ is nothing more than the first order approximation of the N matrix evaluated one-half computation cycle behind. The obvious correction is simply to evaluate the N matrix using Equation (A-58).

$$(A-58) \quad \Theta = \omega_e \left(t - \frac{T}{2} \right)$$

Appendix B

PREPROCESSOR FILTER DERIVATION

The following derivation of a recursive exponentially weighted least squares filter is similar to that for classic least squares. Figure B-1 illustrates fitting a linear function of time to redundant measurements, where

X^* is the measurement,

\hat{X}_n is the estimate of x based on n measurements,

$\hat{\dot{X}}_n$ is the estimate of \dot{x} based on n measurements,

T is the measurement interval.

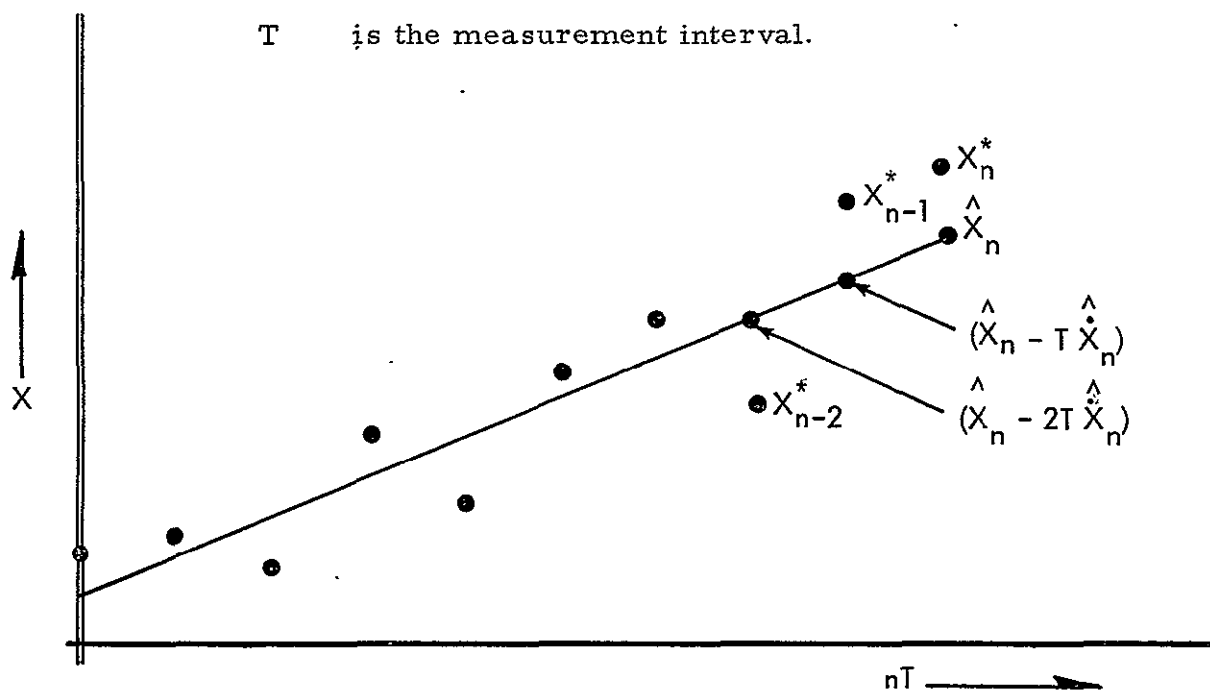


Figure B-1 End Point Linear Fit

Note that the estimates are for the end point (latest time). Solution of the curve fitting process by classic least squares is obtained by minimizing Equation (B-1), which is the sum of the squares of the residuals of all measurements.

$$(B-1) \quad \epsilon^2 = \sum_{i=0}^n (X_{n-i}^* - (\hat{X}_n - i T \hat{\dot{X}}_n))^2$$

This obviously results in an infinite memory filter with equal weighting for all residuals. Thus for classic least squares, a finite memory filter can only be achieved by limiting n which precludes a recursive formulation.

By introducing an exponential weighting factor the infinite memory filter can in effect be reduced to a finite memory filter, without limiting the value of n . For the exponentially weighted case, Equation (B-1) is redefined as

$$(B-2) \quad \epsilon^2 = \sum_{i=0}^n (X_{n-i}^* - (\hat{X}_n - i T \hat{\dot{X}}_n))^2 Z_o^i$$

where

$$(B-3) \quad Z_o^i = (e^{-T/\gamma})^i$$

where γ is the filter time constant. Equation (B-2) places the emphasis on the more recent data by exponentially weighting out the older residuals; which in effect achieves the finite memory. It now remains to derive the recursive formulation.

To minimize Equation (B-2), Equations (B-4) and (B-5) must be satisfied.

$$(B-4) \quad \frac{\partial \epsilon^2}{\partial \hat{X}_n} = \sum_{i=0}^n (X_{n-i}^* - (X_n - i T X_n)) Z_o^i = 0$$

$$(B-5) \quad \frac{\partial \epsilon^2}{\partial \hat{X}_n} = \sum_{i=0}^n i (X_{n-i}^* - (X_n - i T X_n)) Z_o^i = 0$$

Equations (B-4) and (B-5) may be rewritten in matrix form as

$$(B-6) \quad \begin{vmatrix} \sum_{i=0}^n X_{n-i}^* Z_o^i \\ \vdots \\ \sum_{i=0}^n i X_{n-i}^* Z_o^i \end{vmatrix} = \begin{vmatrix} \sum_{i=0}^n Z_o^i & - \sum_{i=0}^n i Z_o^i \\ \vdots & \vdots \\ \sum_{i=0}^n i Z_o^i & - \sum_{i=0}^n i^2 Z_o^i \end{vmatrix} \begin{vmatrix} \hat{X}_n \\ \vdots \\ T \hat{X}_n \end{vmatrix}$$

Using Abel's summation equation, it can be shown that

$$(B-7) \quad a_n = \sum_{i=0}^n Z_o^i = \frac{1 - Z_o^{n+1}}{1 - Z_o}$$

$$(B-8) \quad b_n = \sum_{i=0}^n i Z_o^i = \frac{Z_o}{(1 - Z_o)^2} (1 - (n+1) Z_o^n + n Z_o^{n+1})$$

$$(B-9) \quad c_n = \sum_{i=0}^n i^2 Z_o^i = \frac{Z_o}{(1 - Z_o)^3} (1 + Z_o - Z_o^n ((n+1)^2 - (2n^2 + 2n - 1) Z_o + n^2 Z_o^2))$$

With the above definitions, Equation (B-6) may be rewritten as

$$(B-10) \quad \begin{vmatrix} \sum_{i=0}^n X_{n-i}^* Z_o^i \\ \vdots \\ \sum_{i=0}^n i X_{n-i}^* Z_o^i \end{vmatrix} = \begin{vmatrix} a_n & -b_n \\ \vdots & \vdots \\ b_n & -c_n \end{vmatrix} \begin{vmatrix} \hat{X}_n \\ \vdots \\ T \hat{X}_n \end{vmatrix}$$

Solving for the estimates

$$(B-11) \quad \begin{vmatrix} \hat{X}_n \\ T\hat{X}_n \end{vmatrix} = \begin{vmatrix} a_n & -b_n \\ b_n & -c_n \end{vmatrix}^{-1} \begin{vmatrix} \sum_{i=0}^n X_{n-i}^* Z_o^i \\ \sum_{i=0}^n i X_{n-i}^* Z_o^i \end{vmatrix} = \begin{vmatrix} A_n & B_n \\ -B_n & C_n \end{vmatrix} \begin{vmatrix} \sum_{i=0}^n X_{n-i}^* Z_o^i \\ \sum_{i=0}^n i X_{n-i}^* Z_o^i \end{vmatrix}$$

With proper algebraic manipulation, it can be shown that

$$(B-12) \quad A_n = \frac{(1-Z_o^2) - Z_o^n ((n+1)^2 - n(3n+4)Z_o + (3n^2+2n-1)Z_o^2 - n^2Z_o^3)}{1 - Z_o^n ((n+1)^2 - 2n(n+2)Z_o + (n+1)^2Z_o^2) + Z_o^{2(n+1)}}$$

$$(B-13) \quad B_n = \frac{-(1-Z_o)^2 + Z_o^n ((n+1) - (3n+2)Z_o + (3n+1)Z_o^2 - nZ_o^3)}{1 - Z_o^n ((n+1)^2 - 2n(n+2)Z_o + (n+1)^2Z_o^2) + Z_o^{2(n+1)}}$$

$$(B-14) \quad C_n = \frac{-(1-Z_o)^3 + Z_o^{n+1}(1-Z_o^3)}{Z_o (1 - Z_o^n ((n+1)^2 - 2n(n+2)Z_o + (n+1)^2Z_o^2) + Z_o^{2(n+1)})}$$

Before proceeding further, the summations in Equation (B-11) must be eliminated. Rewriting them in terms of their past values.

$$(B-15) \quad \sum_{i=0}^n X_{n-i}^* Z_o^i = X_n^* + Z_o \sum_{i=0}^{n-1} X_{n-1-i}^* Z_o^i$$

$$(B-16) \quad \sum_{i=0}^n i X_{n-i}^* Z_o^i = Z_o \left(\sum_{i=0}^{n-1} X_{n-1-i}^* Z_o^i + \sum_{i=0}^{n-1} i X_{n-1-i}^* Z_o^i \right)$$

Substituting Equations (B-15) and (B-16) into (B-11)

$$(B-17) \quad \begin{vmatrix} \hat{X}_n \\ T\hat{X}_n \end{vmatrix} = Z_o \begin{vmatrix} (A_n + B_n) & B_n \\ (C_n - B_n) & C_n \end{vmatrix} \begin{vmatrix} \sum_{i=0}^{n-1} X_{n-1-i}^* Z_o^i \\ \sum_{i=0}^{n-1} i X_{n-1-i}^* Z_o^i \end{vmatrix} + \begin{vmatrix} A_n \\ -B_n \end{vmatrix} X_n^*$$

and from Equation (B-10)

$$(B-18) \quad \begin{vmatrix} \sum_{i=0}^{n-1} X_{n-1-i}^* Z_o^i \\ \sum_{i=0}^{n-1} i X_{n-1-i}^* Z_o^i \end{vmatrix} = \begin{vmatrix} a_{n-1} & -b_{n-1} \\ b_{n-1} & -c_{n-1} \end{vmatrix} \begin{vmatrix} \hat{X}_{n-1} \\ T\hat{X}_{n-1} \end{vmatrix}$$

Combining Equations (B-17) and (B-18)

$$(B-19) \quad \begin{vmatrix} \hat{X}_n \\ T\hat{X}_n \end{vmatrix} = Z_o \begin{vmatrix} \frac{a_{n-1}(A_n+B_n) + b_{n-1}B_n}{a_{n-1}(C_n-B_n) + b_{n-1}C_n} & \frac{-b_{n-1}(A_n+B_n) - c_{n-1}B_n}{-b_{n-1}(C_n-B_n) - c_{n-1}C_n} \end{vmatrix} \begin{vmatrix} \hat{X}_{n-1} \\ T\hat{X}_{n-1} \end{vmatrix} + \begin{vmatrix} A_n \\ -B_n \end{vmatrix} X_n^*$$

With proper substitutions and manipulations, Equation (B-19) may be reduced

to

$$(B-20) \quad \begin{vmatrix} \hat{X}_n \\ T\hat{X}_n \end{vmatrix} = \begin{vmatrix} (1-K_1) & (1-K_1) \\ -K_2 & (1-K_2) \end{vmatrix} \begin{vmatrix} \hat{X}_{n-1} \\ T\hat{X}_{n-1} \end{vmatrix} + \begin{vmatrix} K_1 \\ K_2 \end{vmatrix} X_n^*$$

where

$$(B-21) \quad K_1 = A_n$$

and

$$(B-22) \quad K_2 = -B_n$$

The above recursive relationship may be rewritten in the following more familiar form.

$$(B-23) \quad \hat{X}_n = X_p + K_1 (X_n^* - X_p)$$

$$(B-24) \quad T\hat{X}_n = T\hat{X}_{n-1} + K_2 (X_n^* - X_p)$$

where

$$(B-25) \quad X_p = \hat{X}_{n-1} + T\hat{X}_{n-1}$$

K_1 and K_2 are functions of n and Z_o , and are restated in Equations (B-26) and (B-27) for the exponentially weighted least squares case.

$$(B-26) \quad K_1 = \frac{(1-Z_o^2) - Z_o^n ((n+1)^2 - n(3n+4)Z_o + (3n^2+2n-1)Z_o^2 - n^2Z_o^3)}{1 - Z_o^n ((n+1)^2 - 2n(n+2)Z_o + (n+1)^2Z_o^2) + Z_o^{2(n+1)}},$$

$$(0 \leq Z_o < 1)$$

$$(B-27) \quad K_2 = \frac{(1-Z_o)^2 - Z_o^n ((n+1) - (3n+2)Z_o + (3n+1)Z_o^2 - nZ_o^3)}{1 - Z_o^n ((n+1)^2 - 2n(n+2)Z_o + (n+1)^2Z_o^2) + Z_o^{2(n+1)}}, \quad (0 \leq Z_o < 1)$$

By evaluating the above expressions for $Z_o = 1$, the gains for classic least squares are obtained. They are

$$(B-28) \quad K_1 \Big|_{Z_o=1} = \frac{2(2n+1)}{(n+1)(n+2)}$$

$$K_2 \Big|_{Z_o=1} = \frac{6}{(n+1)(n+2)}$$

The final consideration is the case for large n , which yields the preprocessor filter discussed in Section 4. As n approaches infinity, Equations (B-26) and (B-27) reduce to

$$(B-30) \quad K_1 \Big|_{n \rightarrow \infty} = 1 - Z_o^2$$

$$(B-31) \quad K_2 \Big|_{n \rightarrow \infty} = (1 - Z_o)^2$$

The rate of convergence of the gains as a function of n , for various Z_o values, are plotted in Figures B-2 and B-3. Convergence requires about four time constants; i. e., the time required to clear the initial transient.

Of more interest are the curves plotted in Figures B-4 through B-7. These show the ratio of the standard deviation of the estimates to the standard deviation of the measurements for Z_o equal to 0.7 and 0.9. As would be expected, under steady-state conditions the results are identical.

While the previous curves indicate the desirability of a long memory filter to reduce to noise power, the value of the finite memory is illustrated in Figures B-8 through B-11. These curves show the error in the estimate resulting from acceleration, which the first order filter is not designed to recognize.

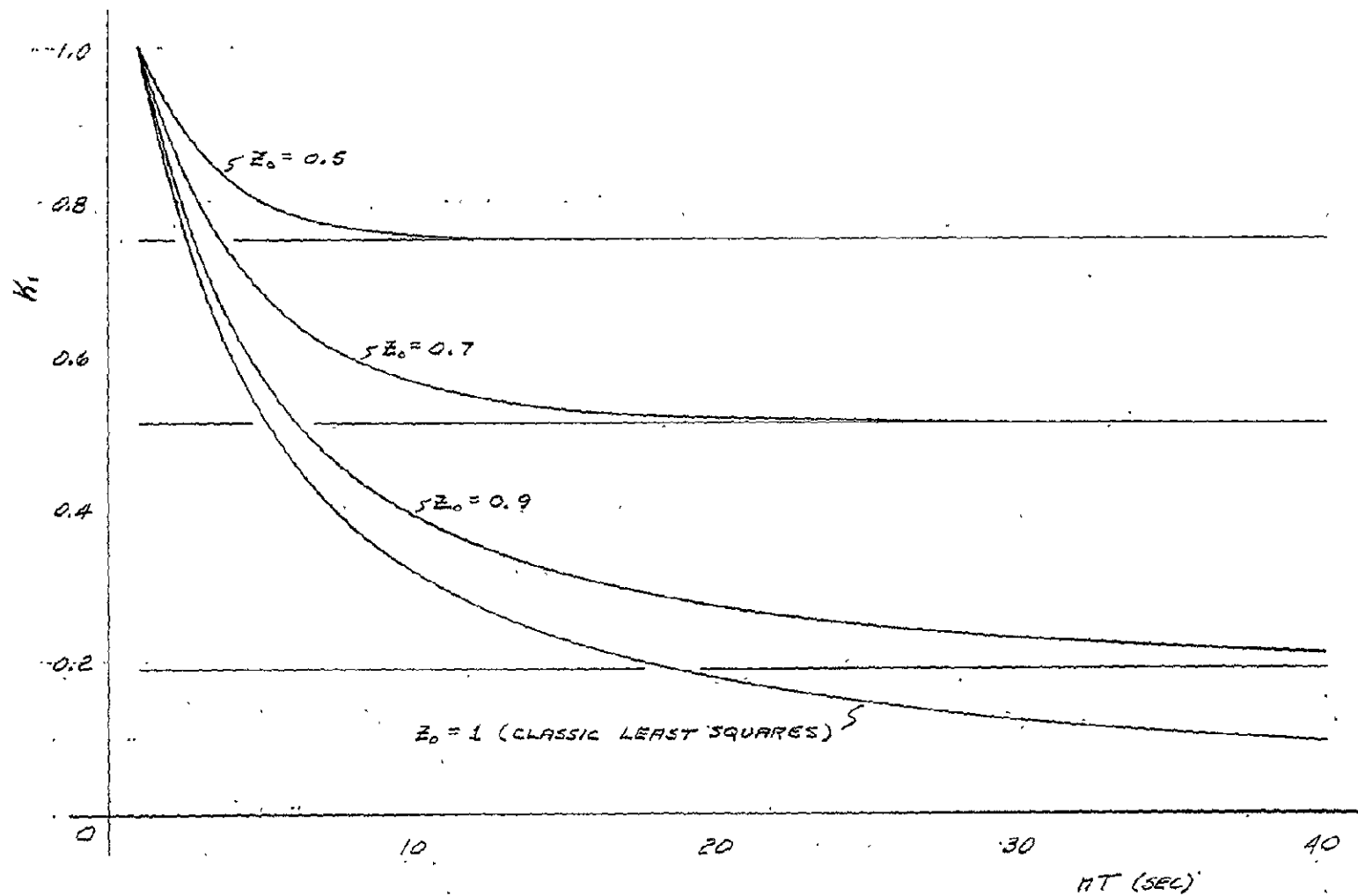


Figure B-2 Exponentially Weighted Least Squares Position Gain

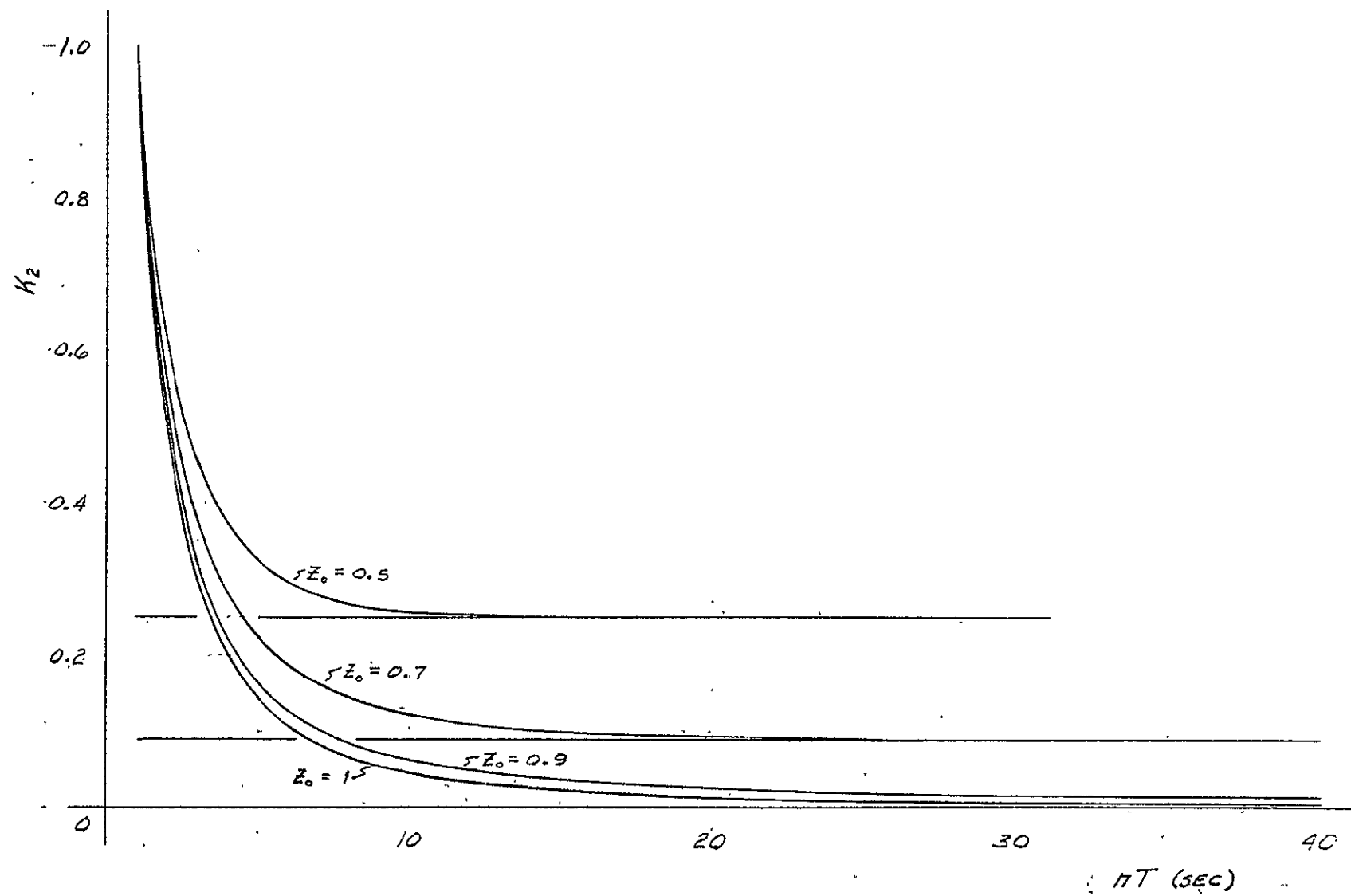


Figure B-3 Exponentially Weighted Least Squares Rate Gain

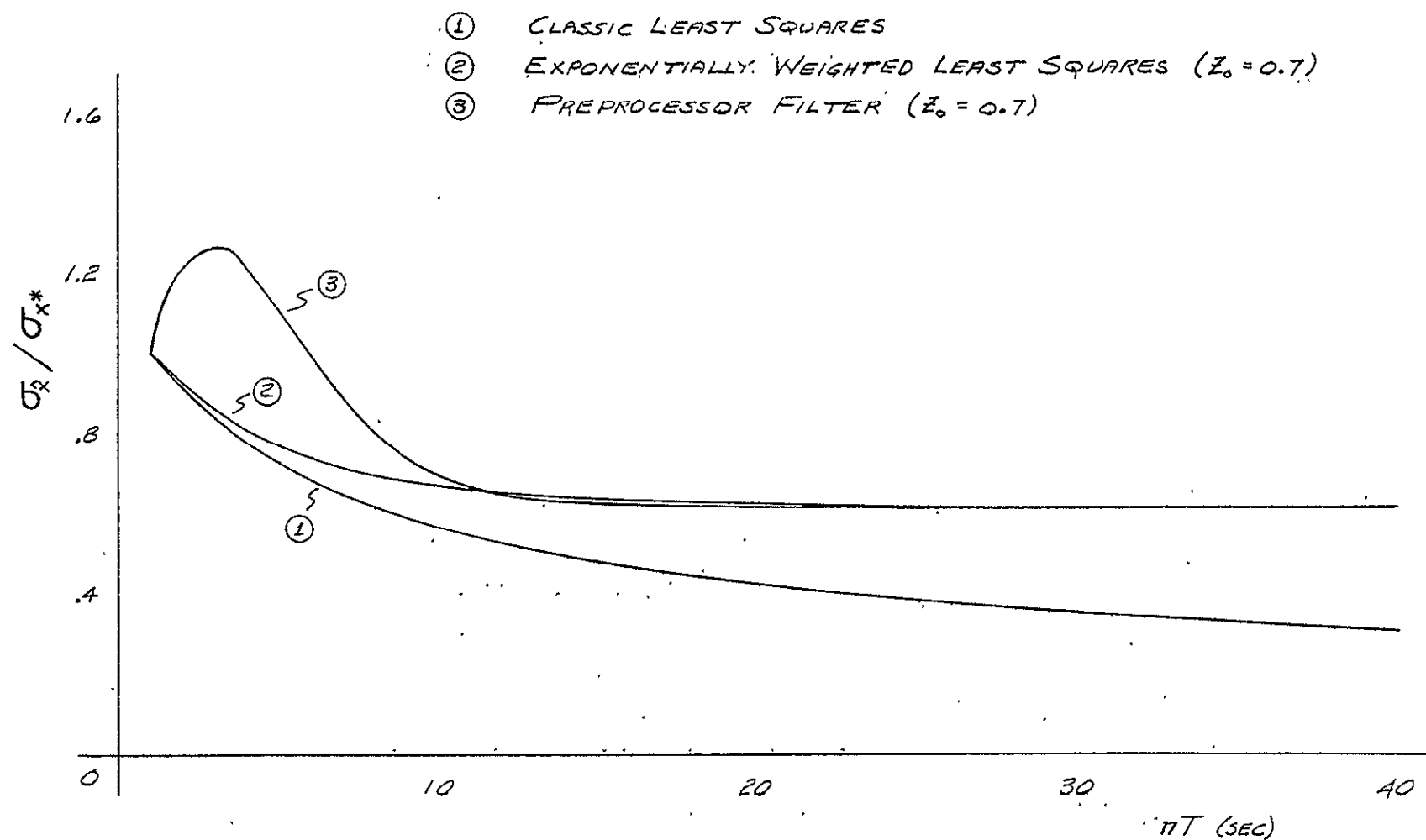


Figure B-4 Ratio of Standard Deviations of Position Estimate to Measurement ($Z_0 = 0.7$)

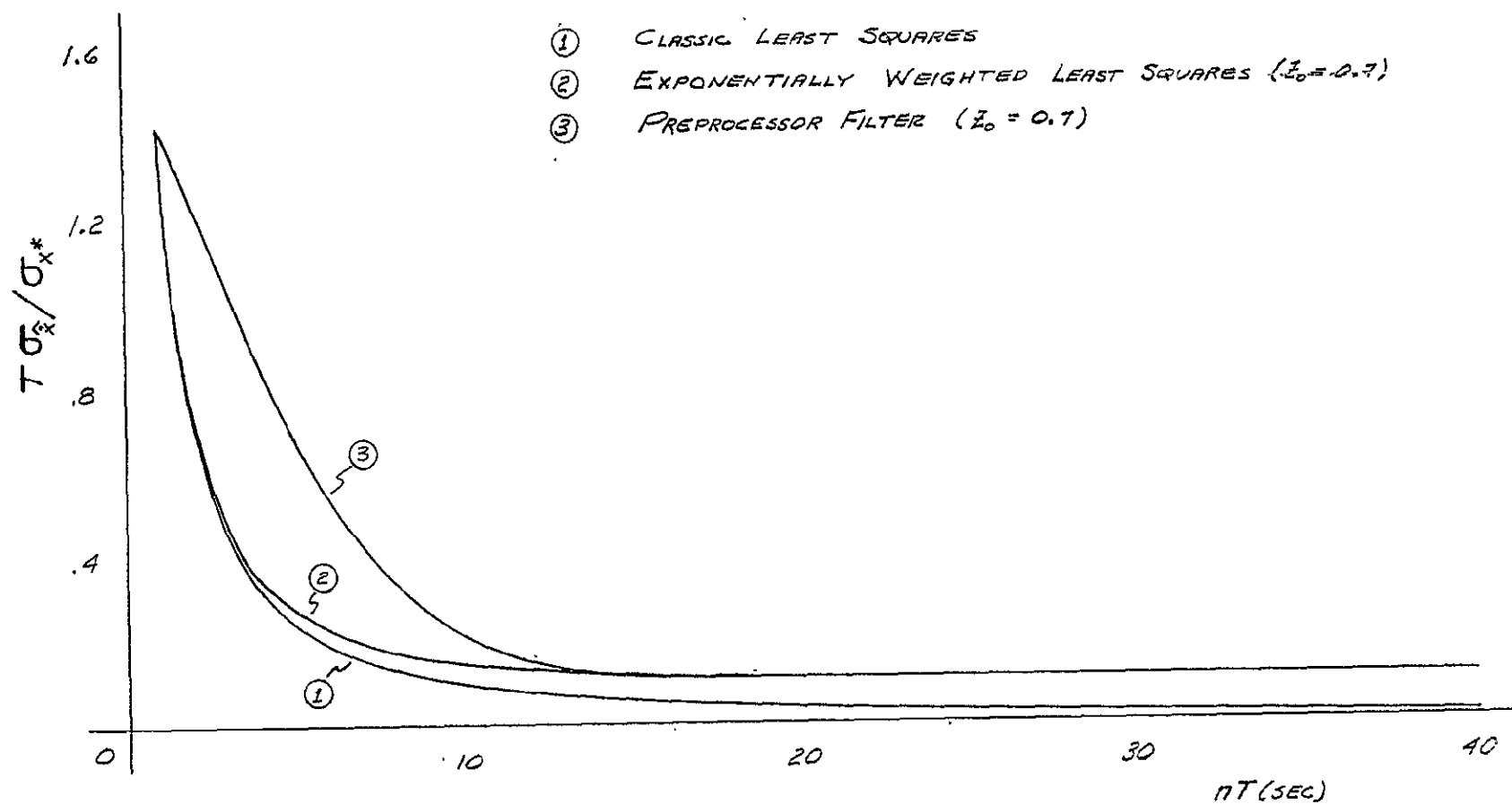


Figure B-5 Ratio of Standard Deviations of Velocity Estimate to Measurement ($Z_0 = 0.7$)

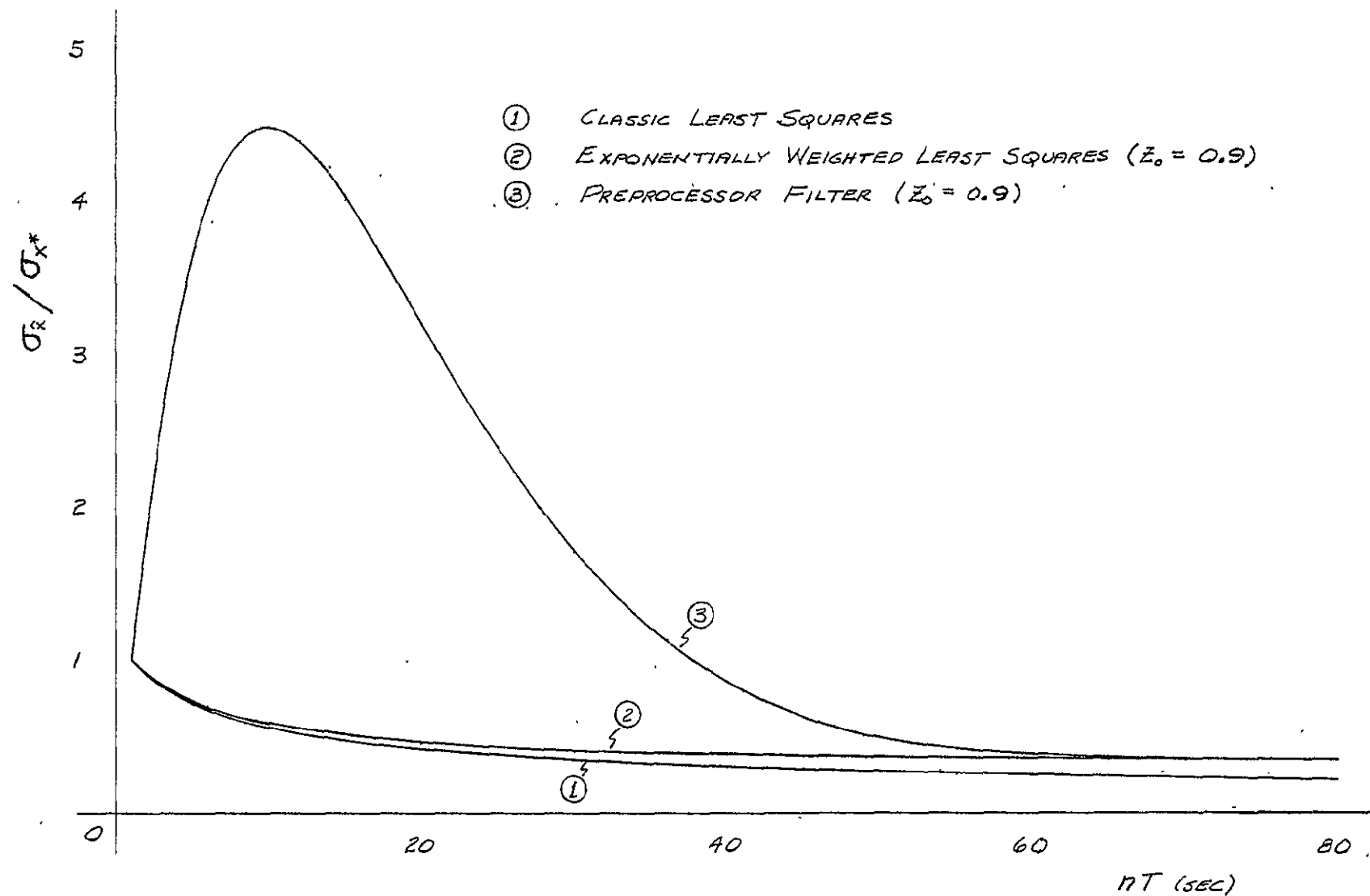


Figure B-6 Ratio of Standard Deviations of Position Estimate to Measurement ($Z_0 = 0.9$)

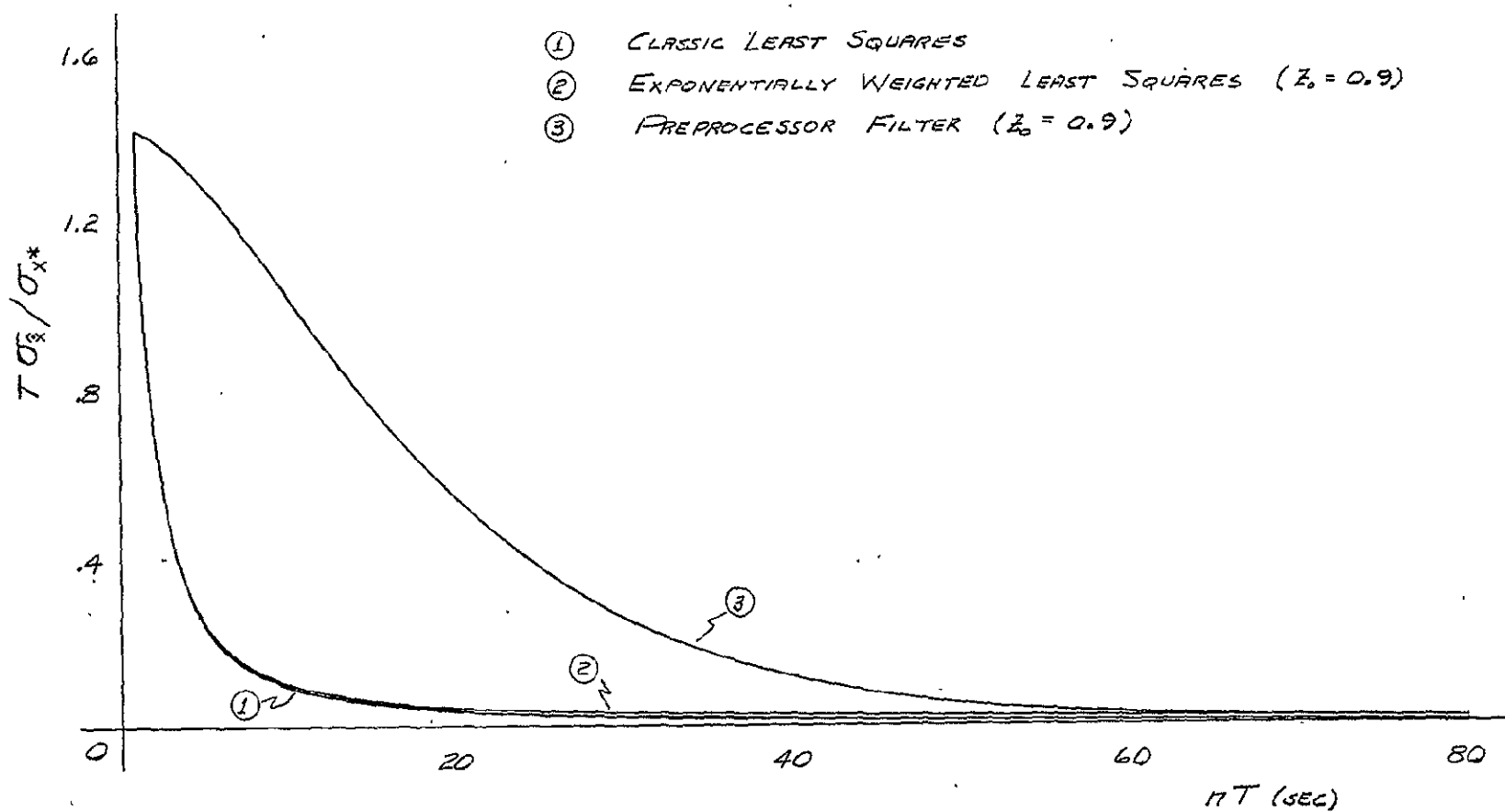


Figure B-7 Ratio of Standard Deviations of Velocity Estimate to Measurement ($Z_0 = 0.9$)

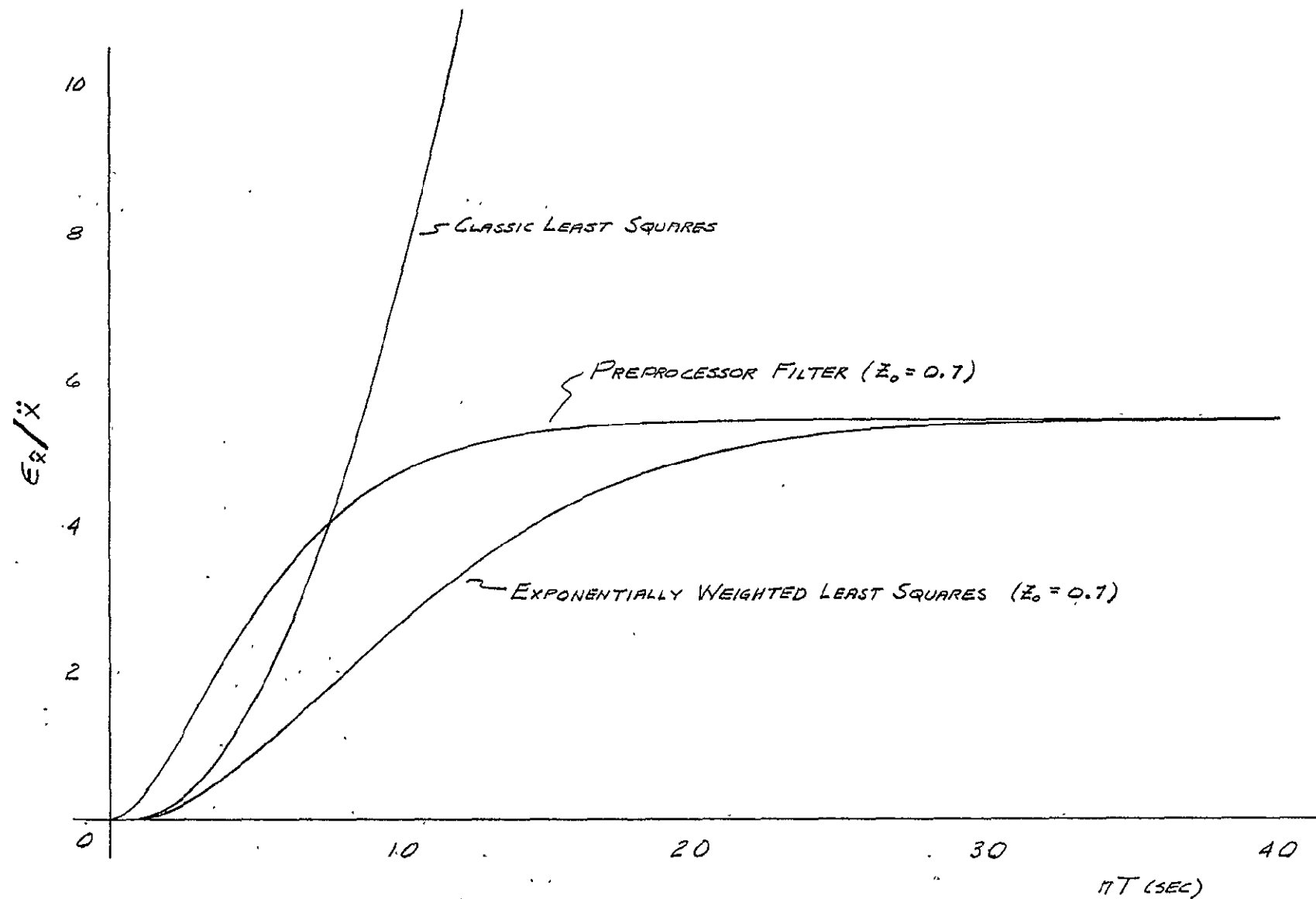


Figure B-8 Error in Position Estimate Due to Constant Acceleration Input ($Z_0 = 0.7$)

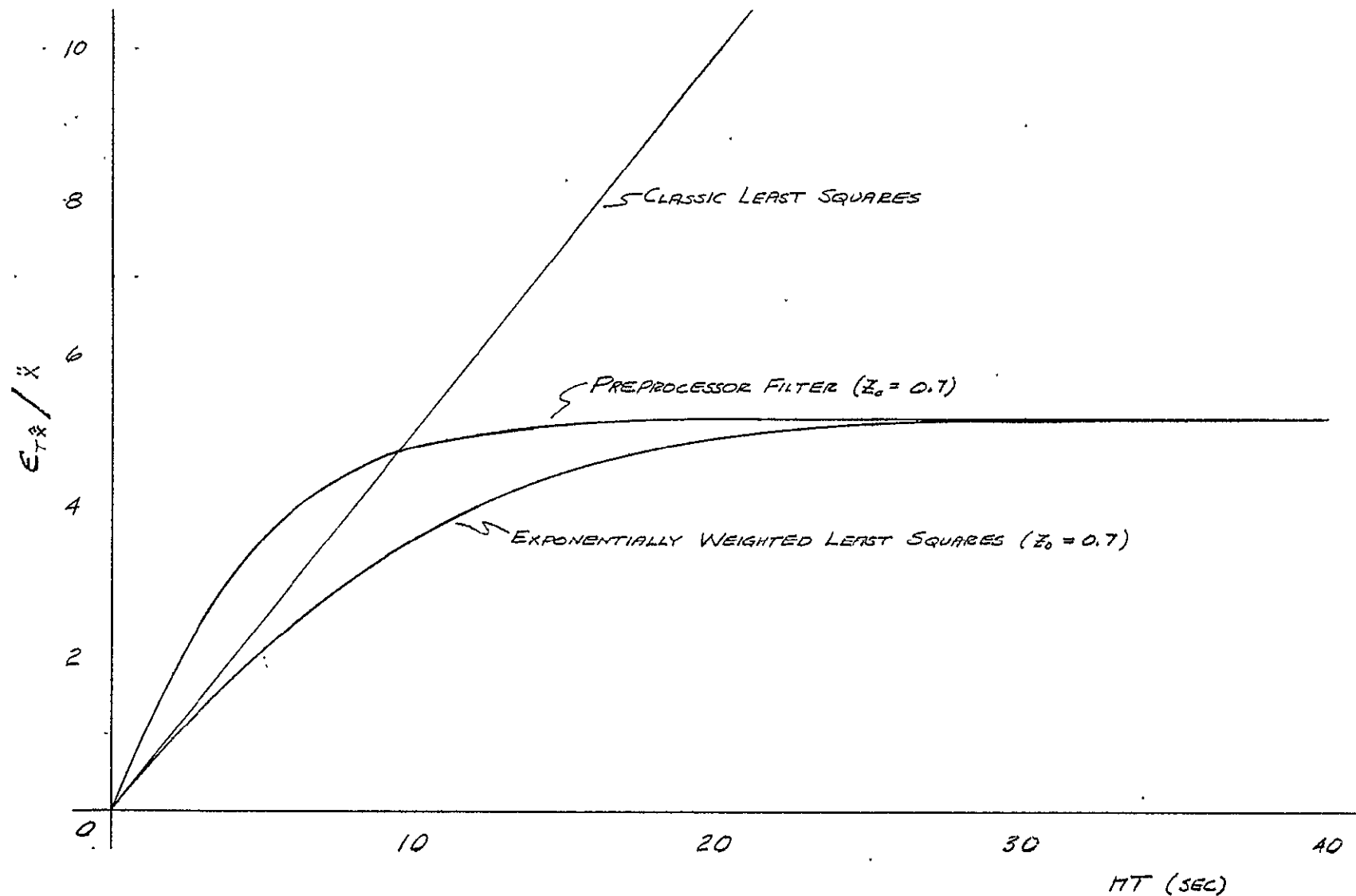


Figure B-9 Error in Velocity Estimate Due to Constant Acceleration Input ($Z_0 = 0.7$)

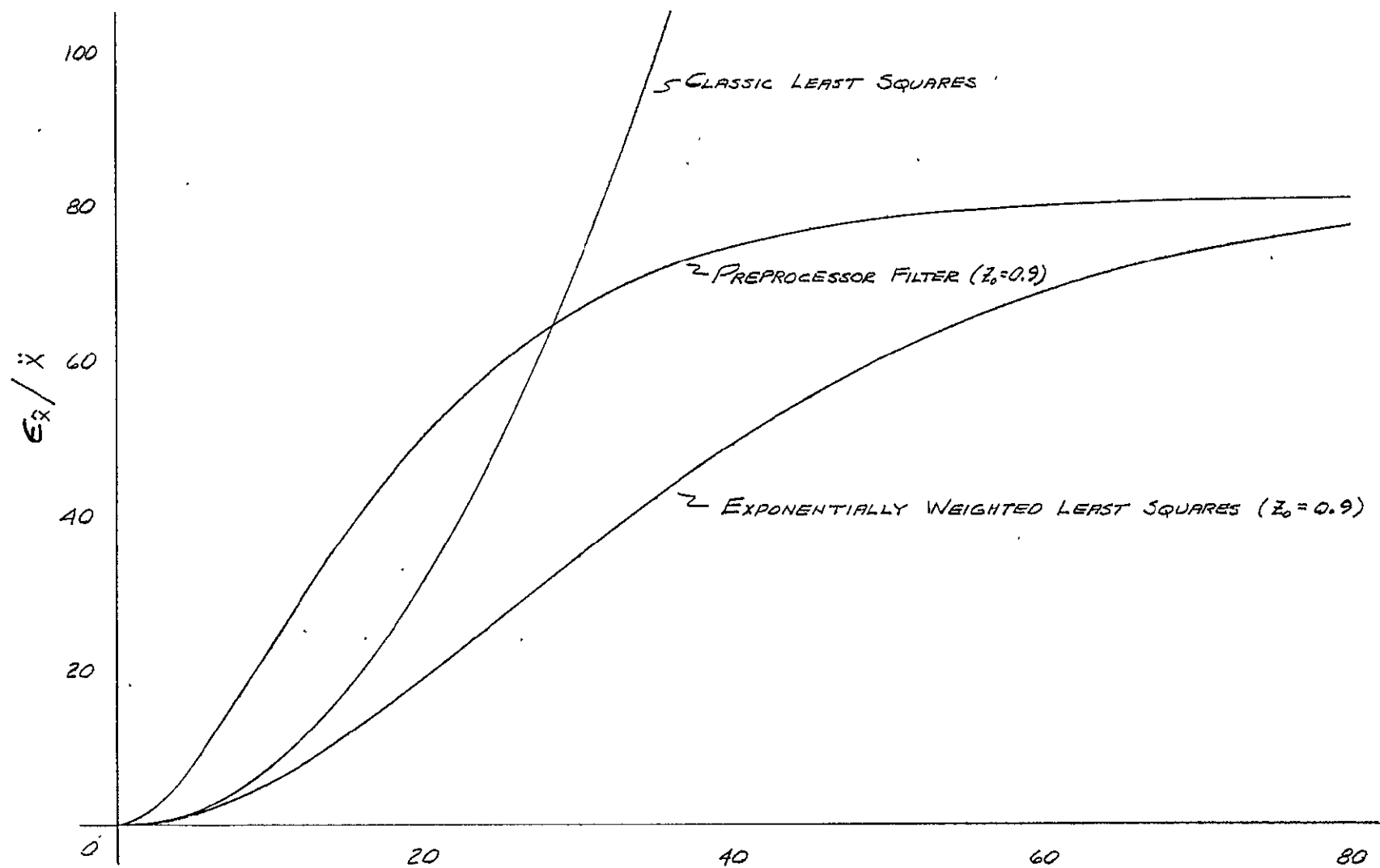


Figure B-10 Error in Position Estimate Due to Constant Acceleration Input ($Z_0 = 0.9$) τT

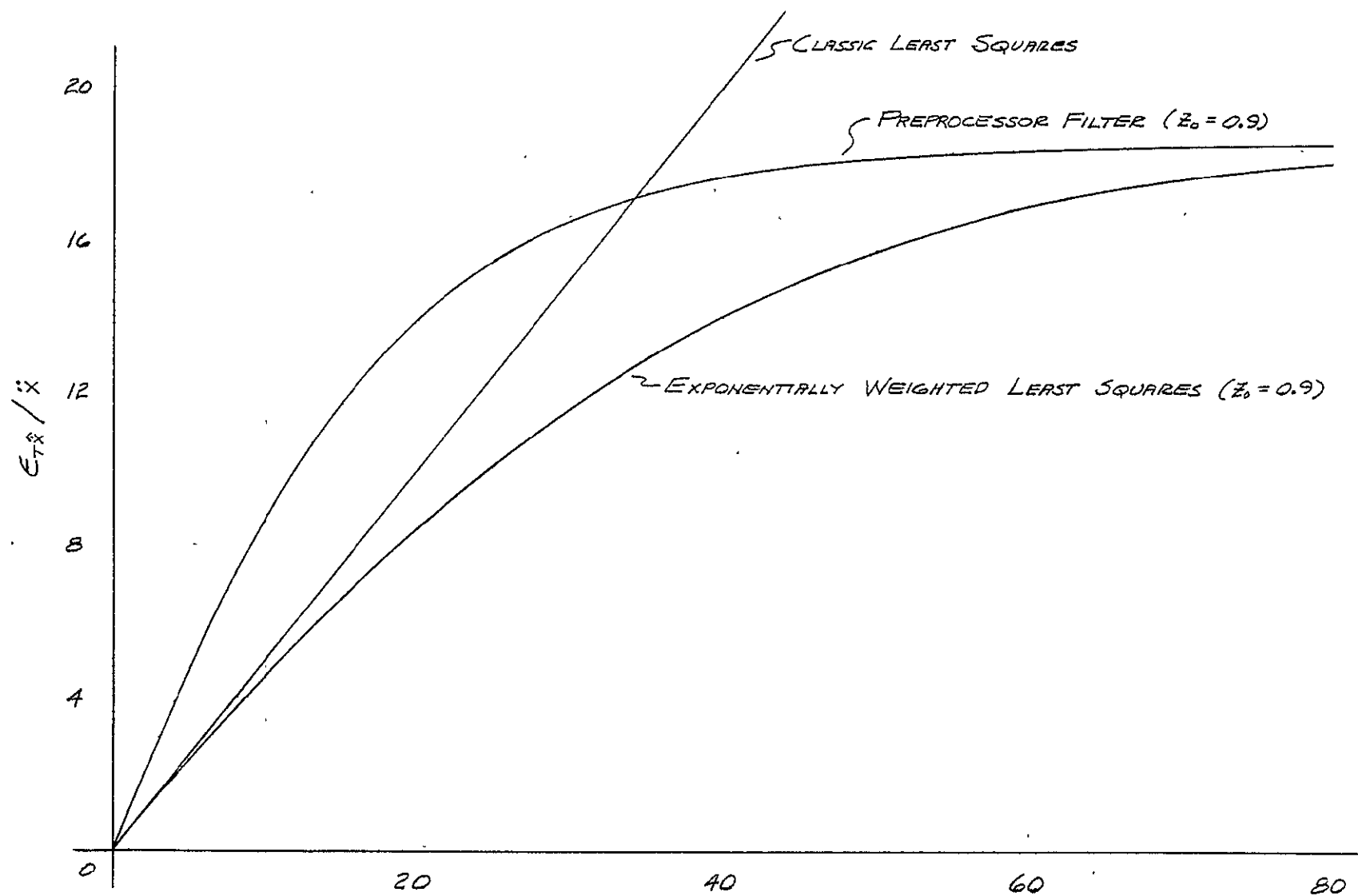


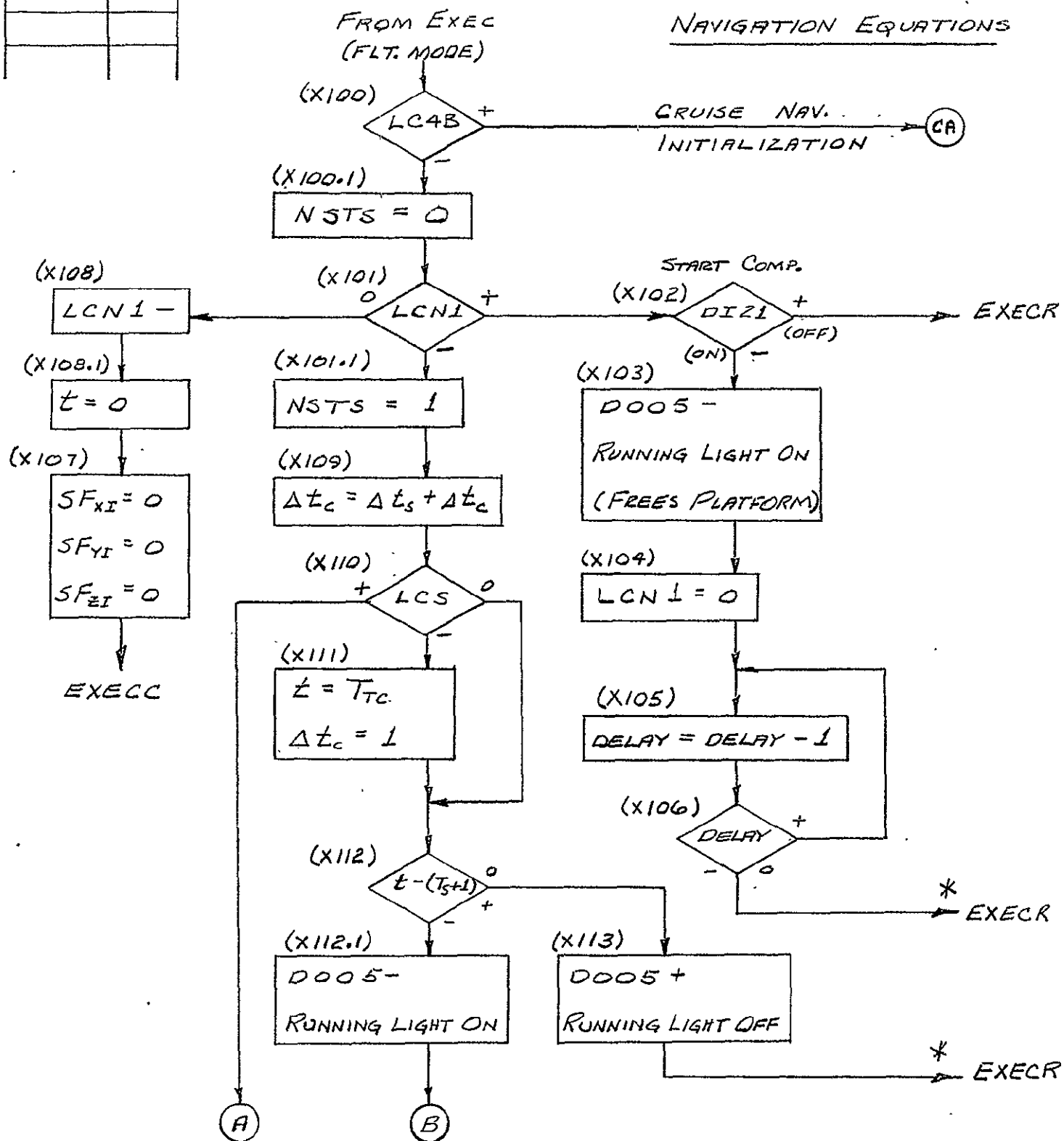
Figure B-11 Error in Velocity Estimate Due to Constant Acceleration Input ($Z_0 = 0.9$)

Appendix C

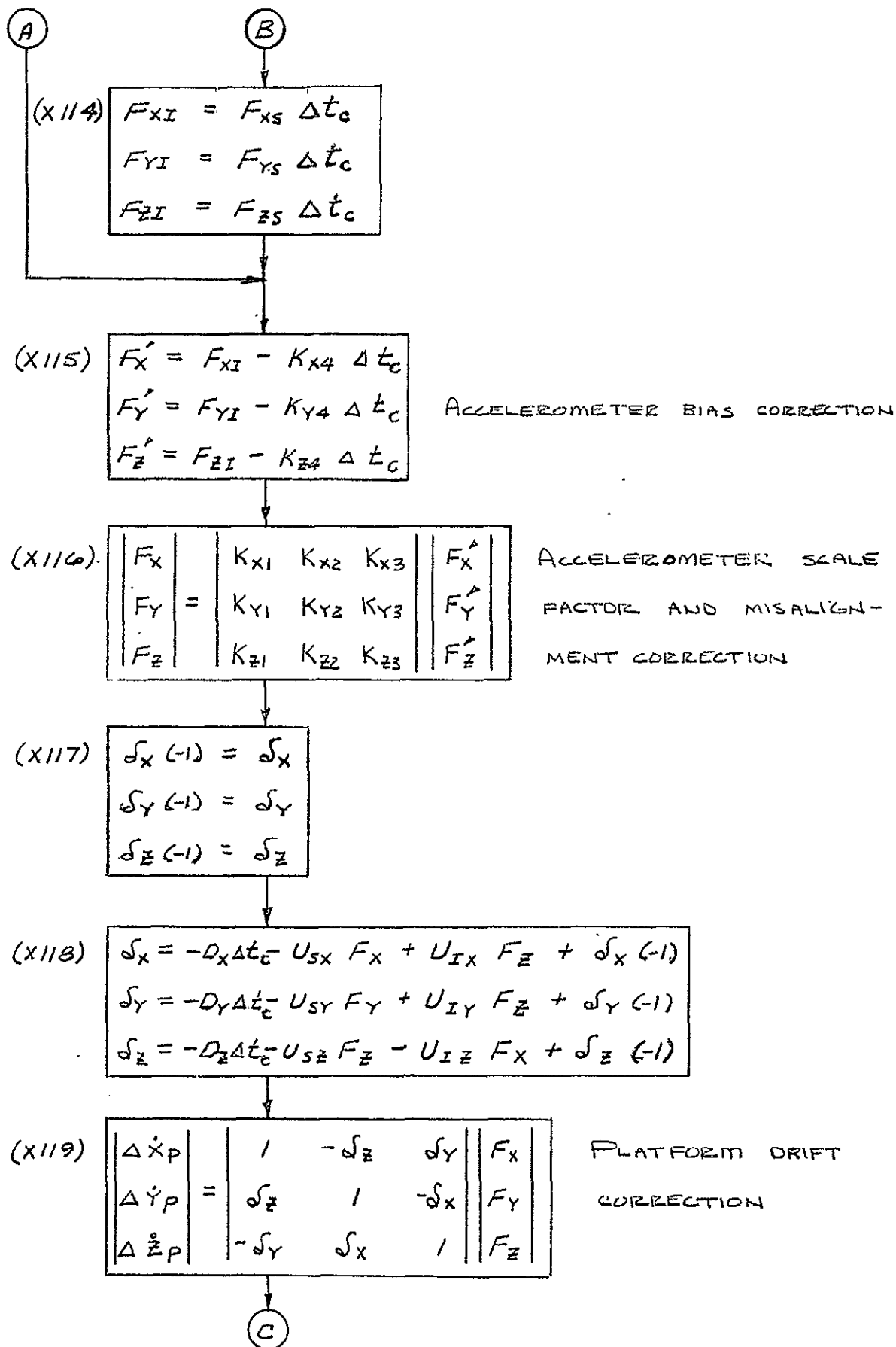
SYSTEM MATH FLOW

The system math flow contained in this Appendix is the final Revision B level for the Phase 1B software. A math flow symbol definition list is contained at the back of the appendix.

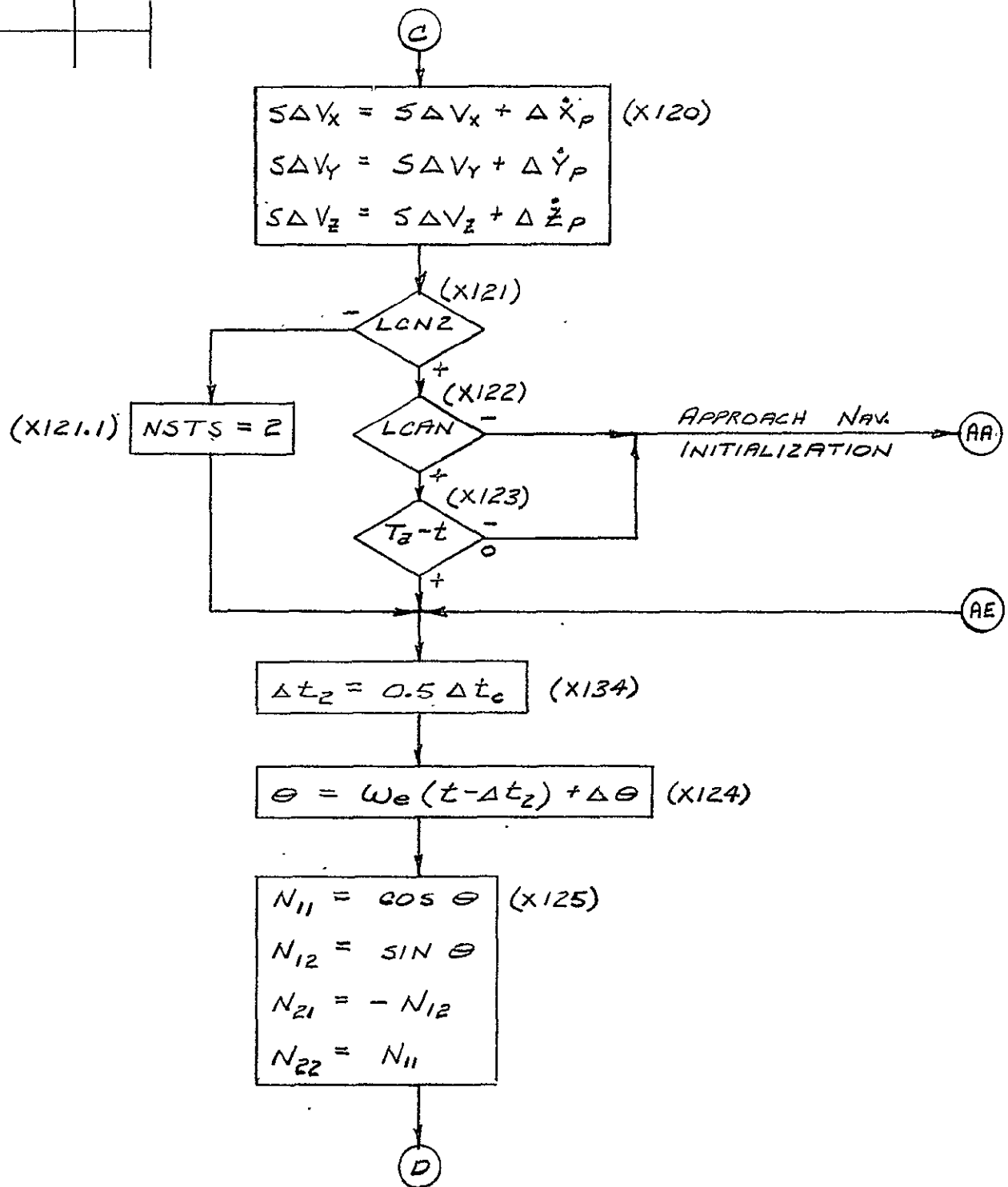
DATE	No.
9-23-68	95



* DO NOT UPDATE TELEMETRY BUFFER



DATE	No.
9-23-68	95



(D)

(X126)

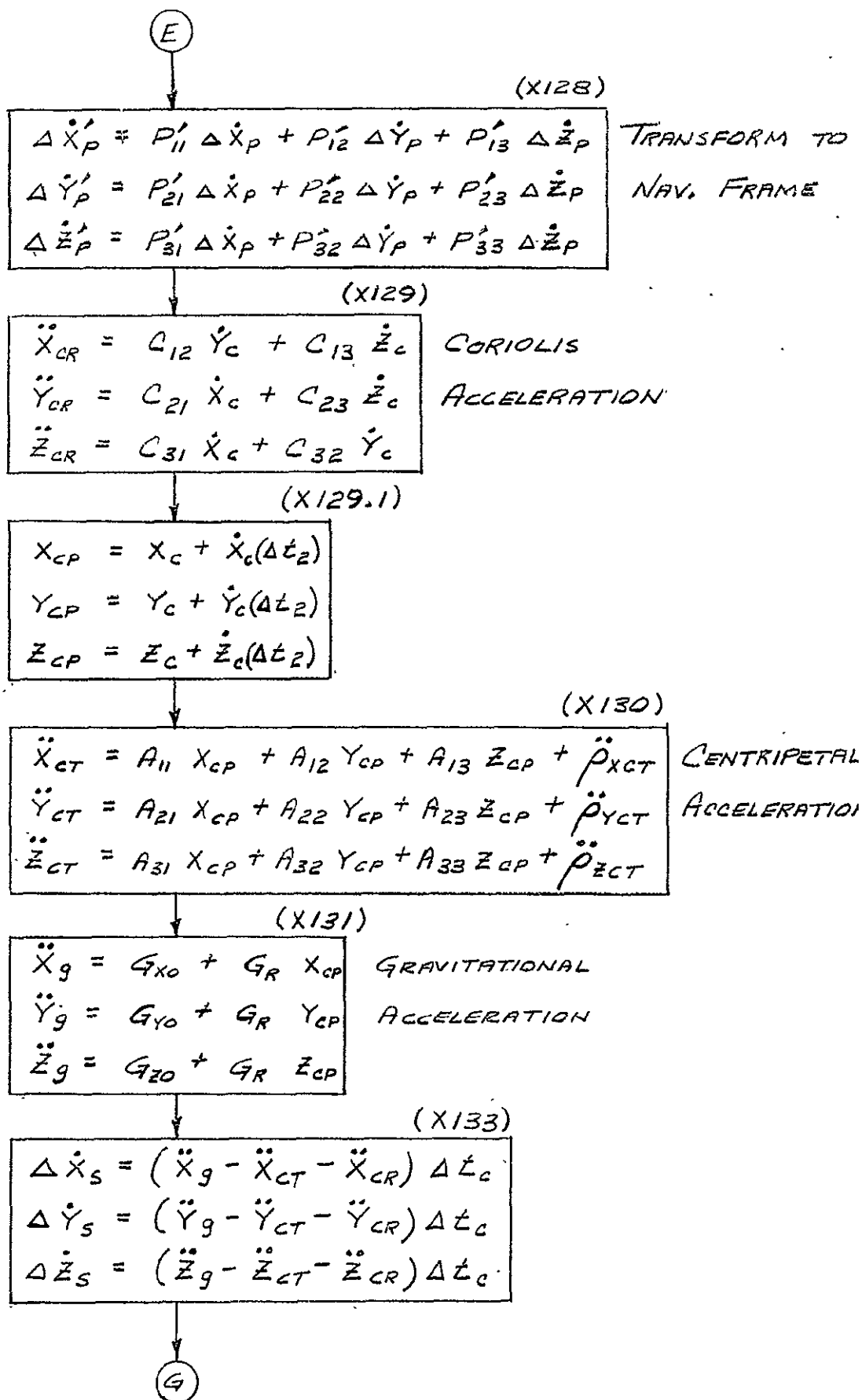
$$\begin{aligned}D'_{11} &= D_{11} N_{11} + D_{12} N_{21} \\D'_{12} &= D_{11} N_{12} + D_{12} N_{22} \\D'_{13} &= D_{13} \\D'_{21} &= D_{21} N_{11} + D_{22} N_{21} \\D'_{22} &= D_{21} N_{12} + D_{22} N_{22} \\D'_{23} &= D_{23} \\D'_{31} &= D_{31} N_{11} \\D'_{32} &= D_{31} N_{12} \\D'_{33} &= D_{33}\end{aligned}$$

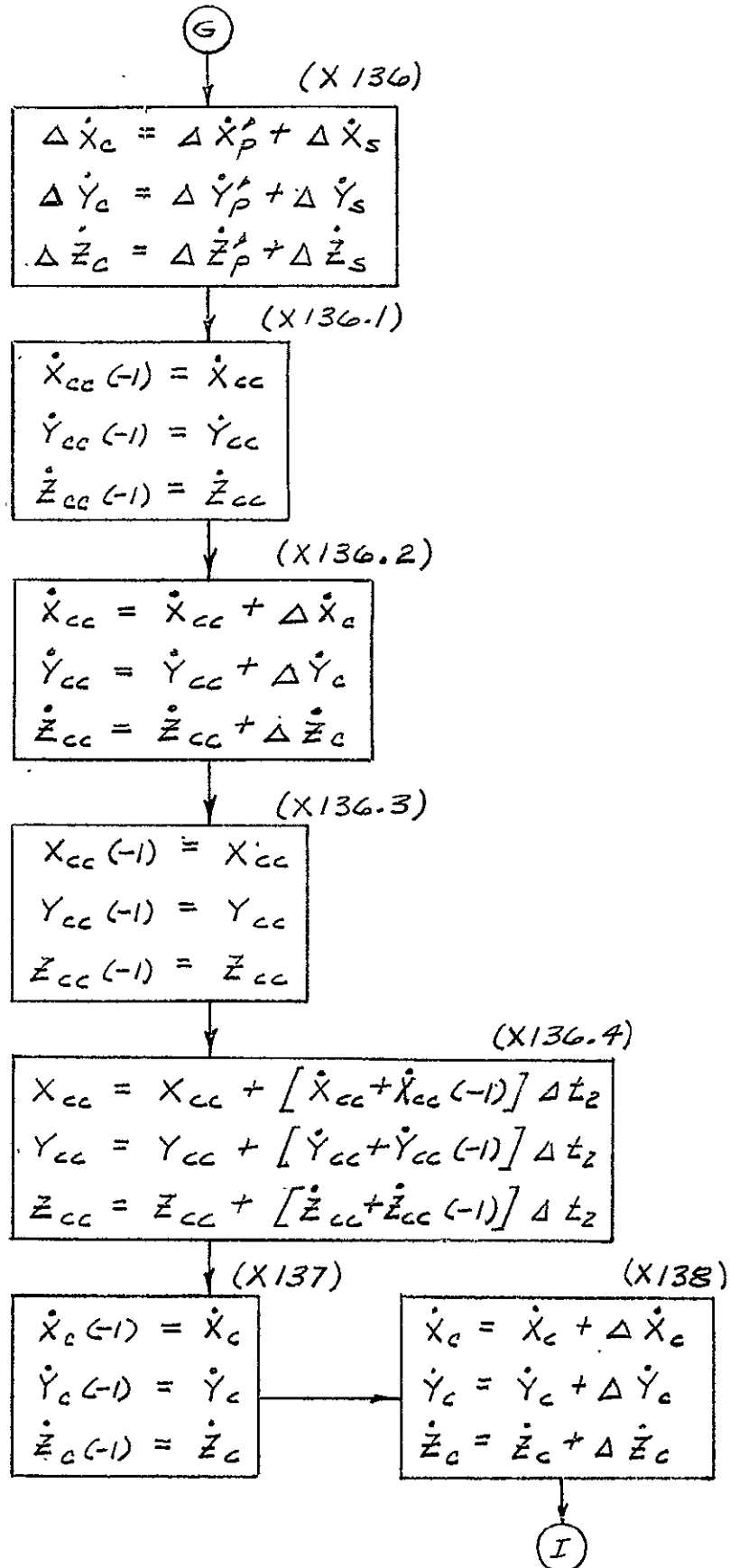
(X127)

$$\begin{aligned}P'_{11} &= D'_{11} P_{11} + D'_{12} P_{21} + D'_{13} P_{31} \\P'_{12} &= D'_{11} P_{12} + D'_{13} P_{32} \\P'_{13} &= D'_{11} P_{13} + D'_{12} P_{23} + D'_{13} P_{33} \\P'_{21} &= D'_{21} P_{11} + D'_{22} P_{21} + D'_{23} P_{31} \\P'_{22} &= D'_{21} P_{12} + D'_{23} P_{32} \\P'_{23} &= D'_{21} P_{13} + D'_{22} P_{23} + D'_{23} P_{33} \\P'_{31} &= D'_{31} P_{11} + D'_{32} P_{21} + D'_{33} P_{31} \\P'_{32} &= D'_{31} P_{12} + D'_{33} P_{32} \\P'_{33} &= D'_{31} P_{13} + D'_{32} P_{23} + D'_{33} P_{33}\end{aligned}$$

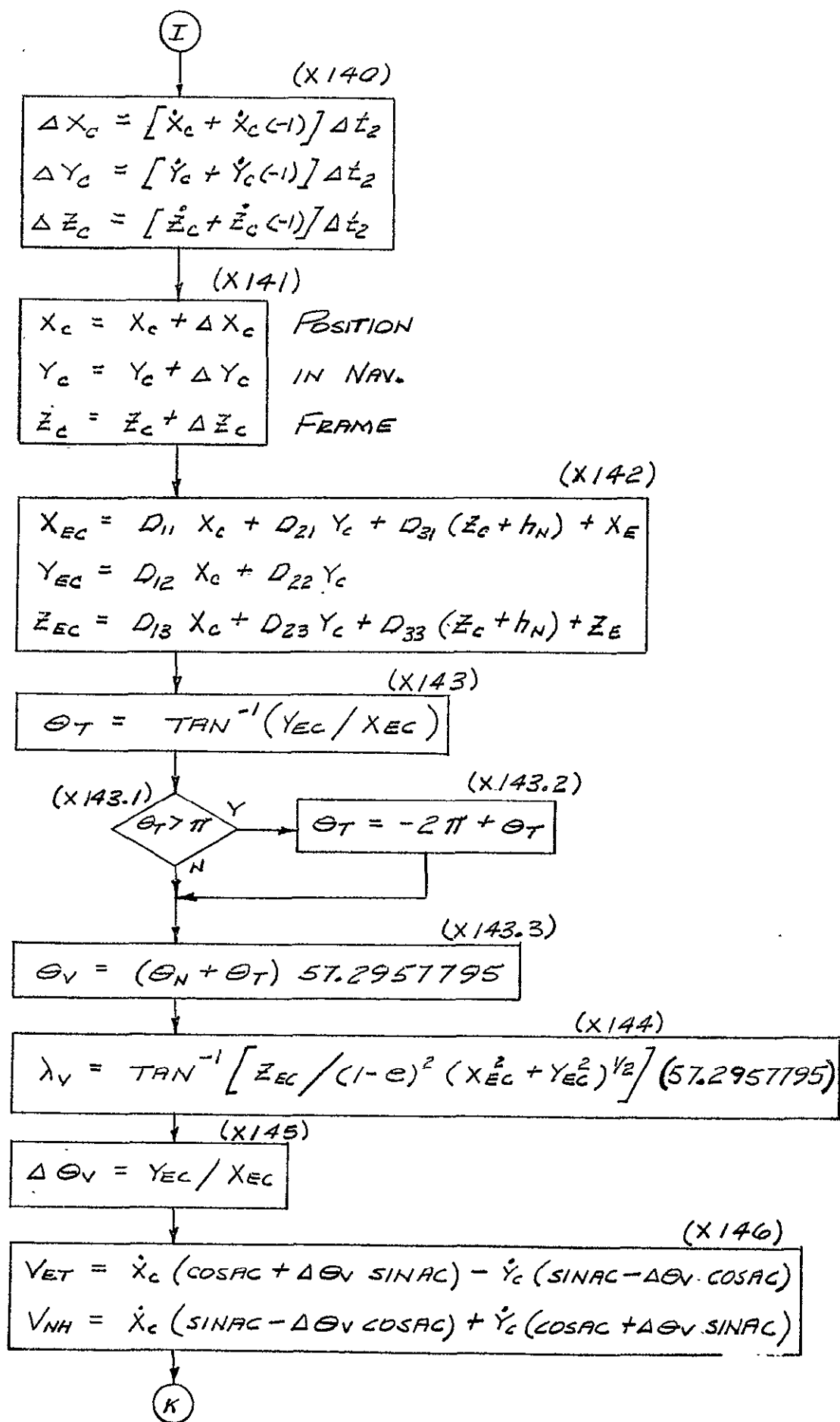
(E)

DATE	No.
9-23-68	95

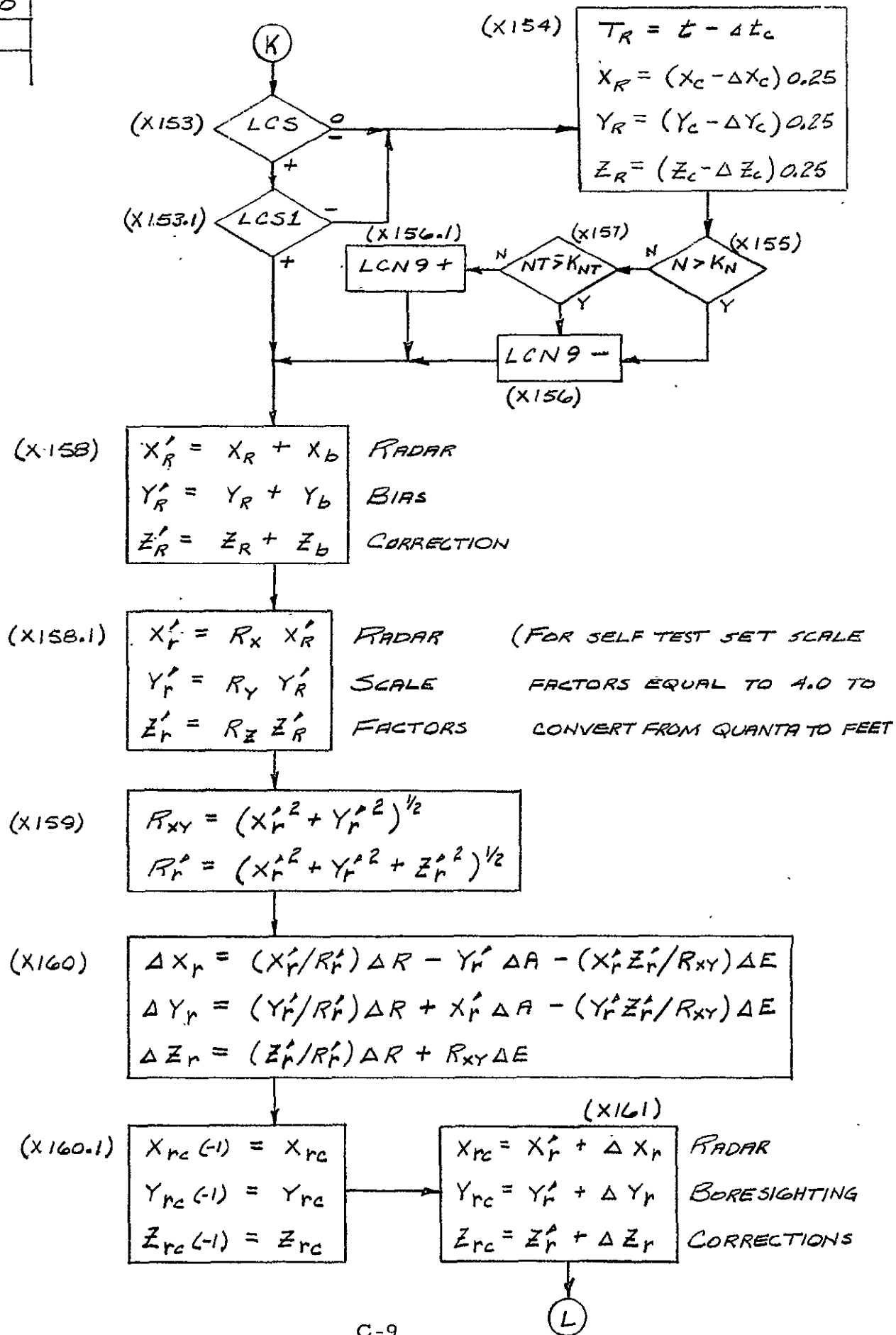




VELOCITY IN
NAV. FRAME

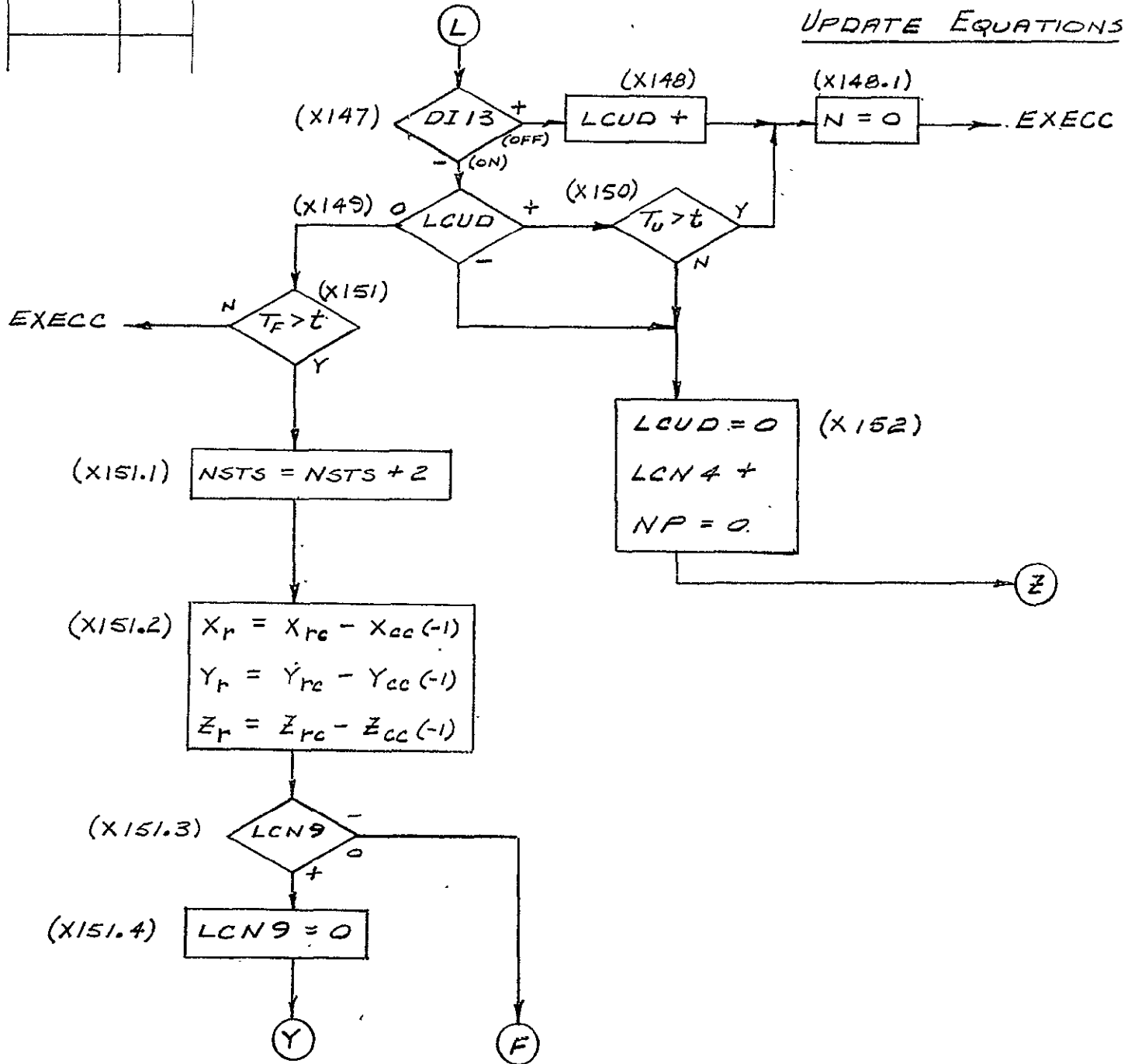


DATE	No.
9-23-68	95
10-7-68	100

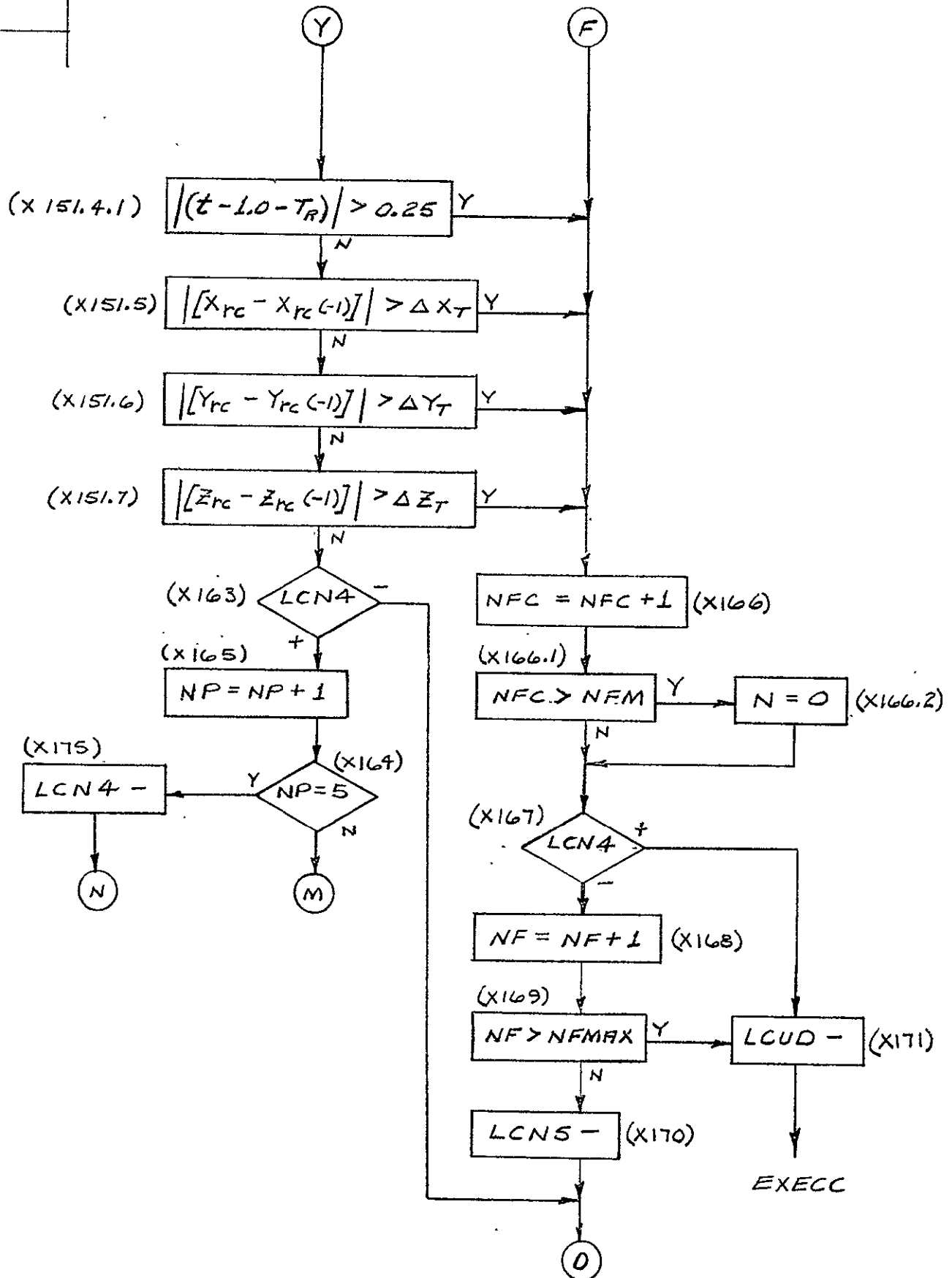


DATE	No.
9-23-68	95
10-7-68	100

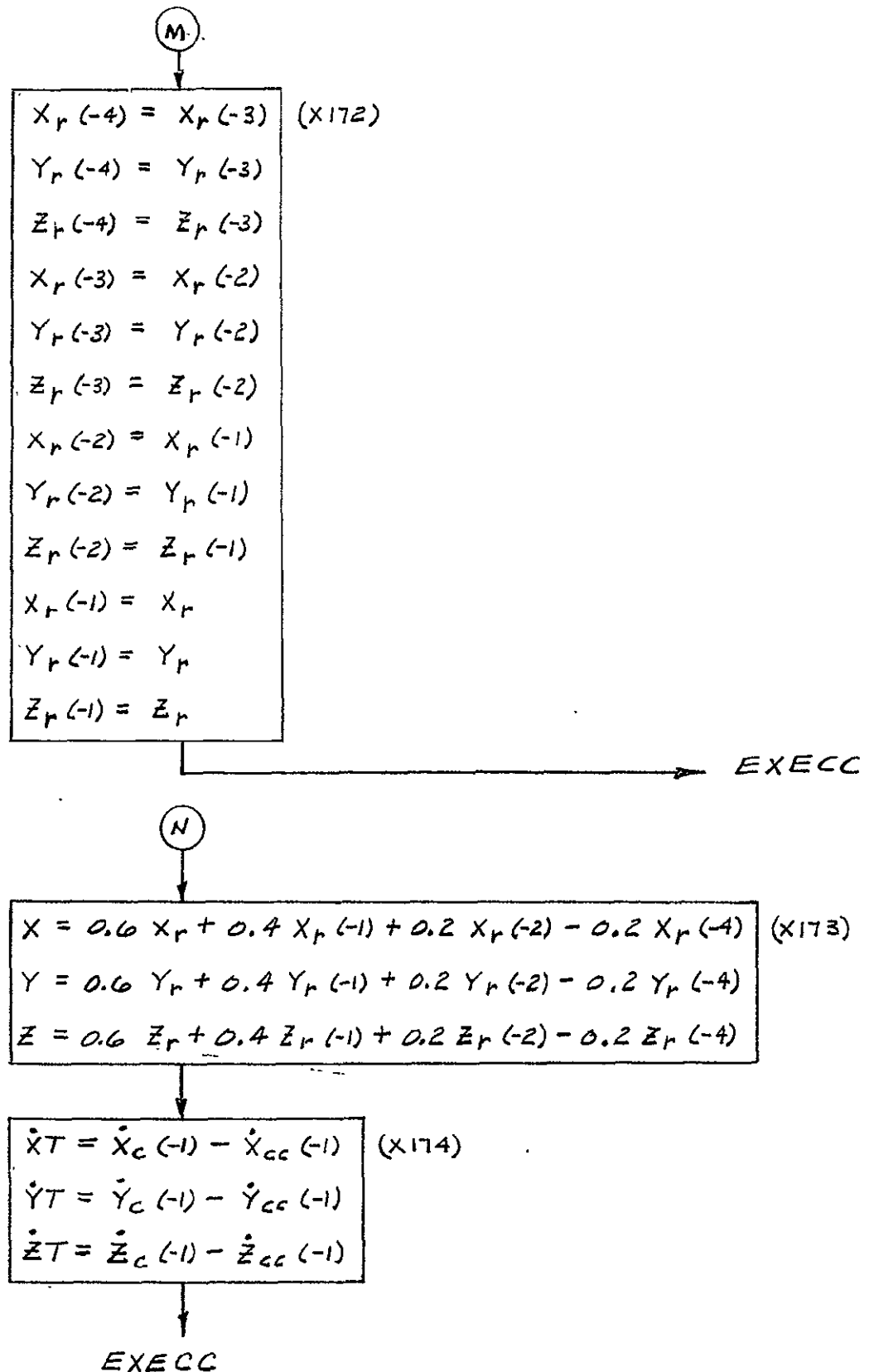
UPDATE EQUATIONS



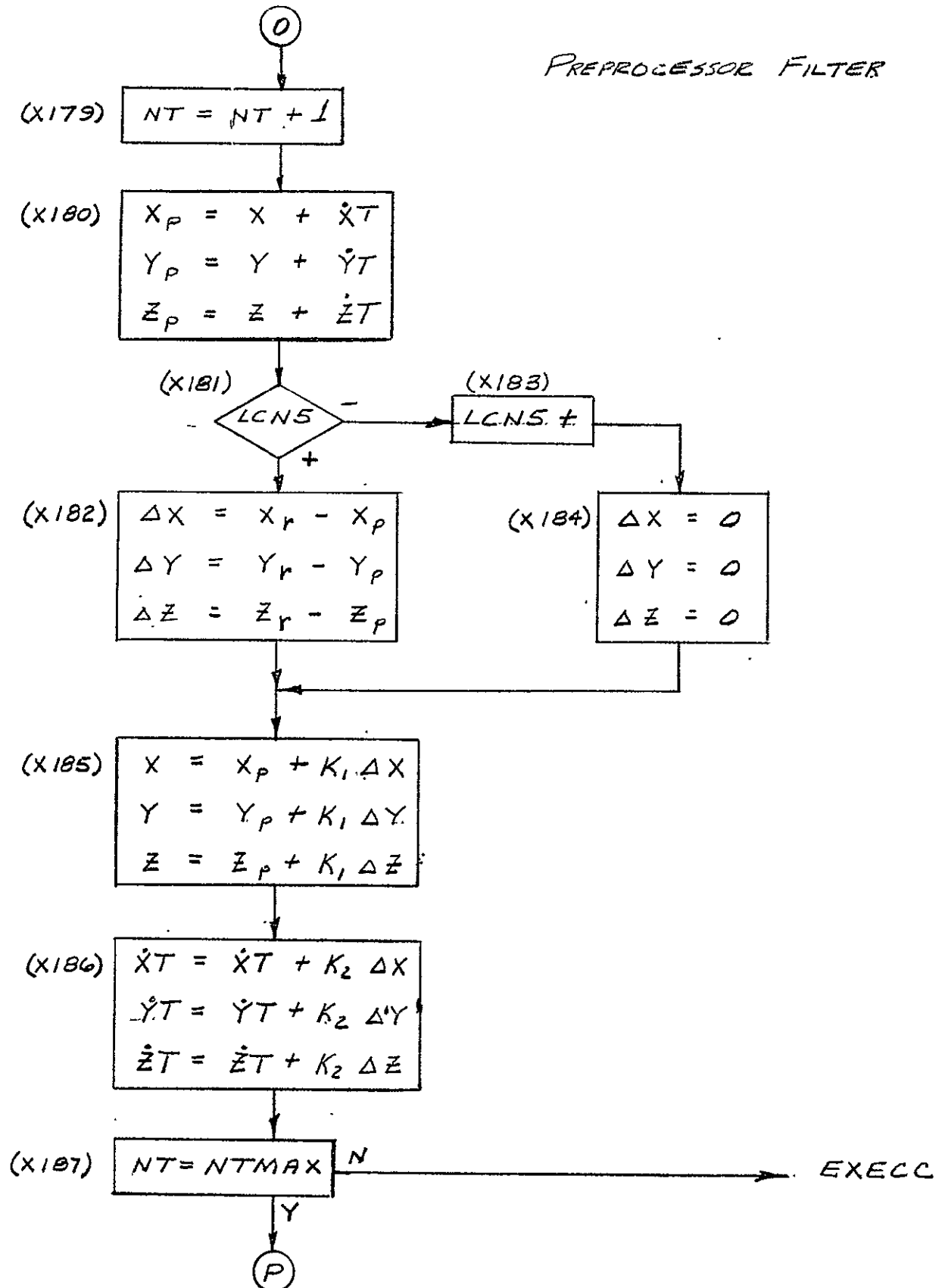
DATE	No.
9-23-68	95
10-7-68	100



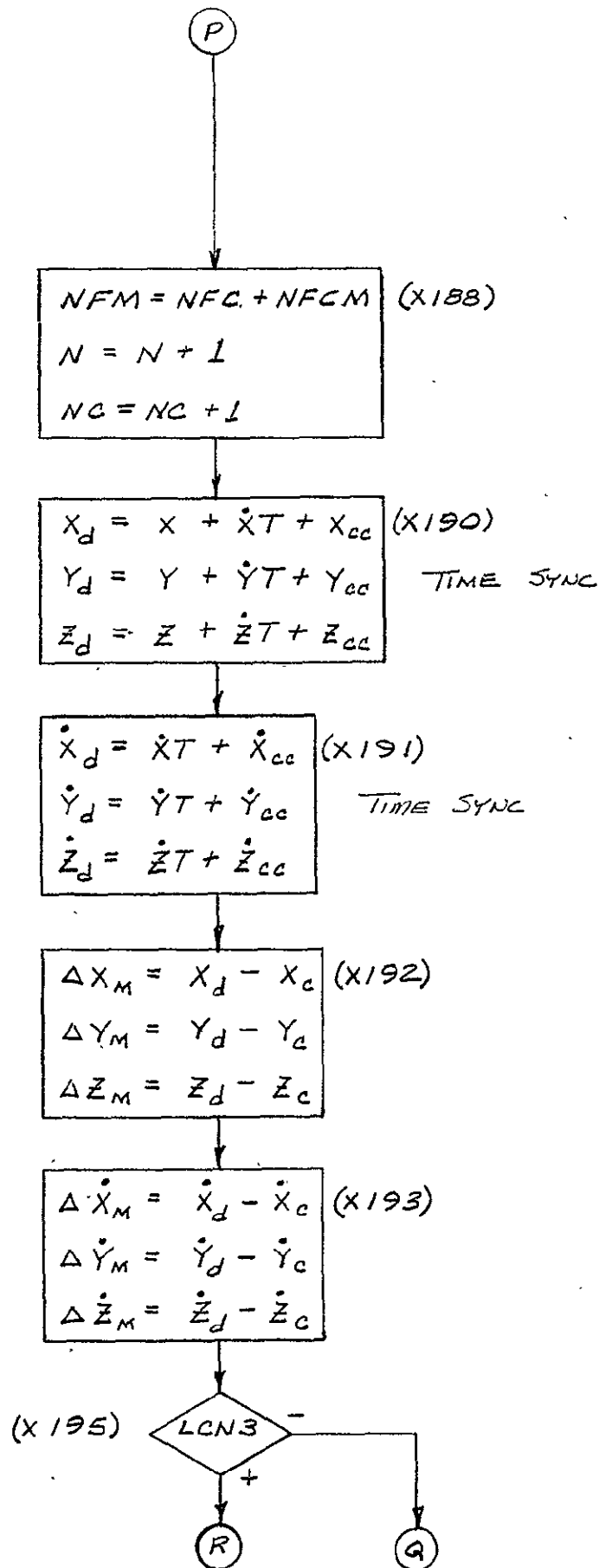
DATE	NO.
9-23-68	95



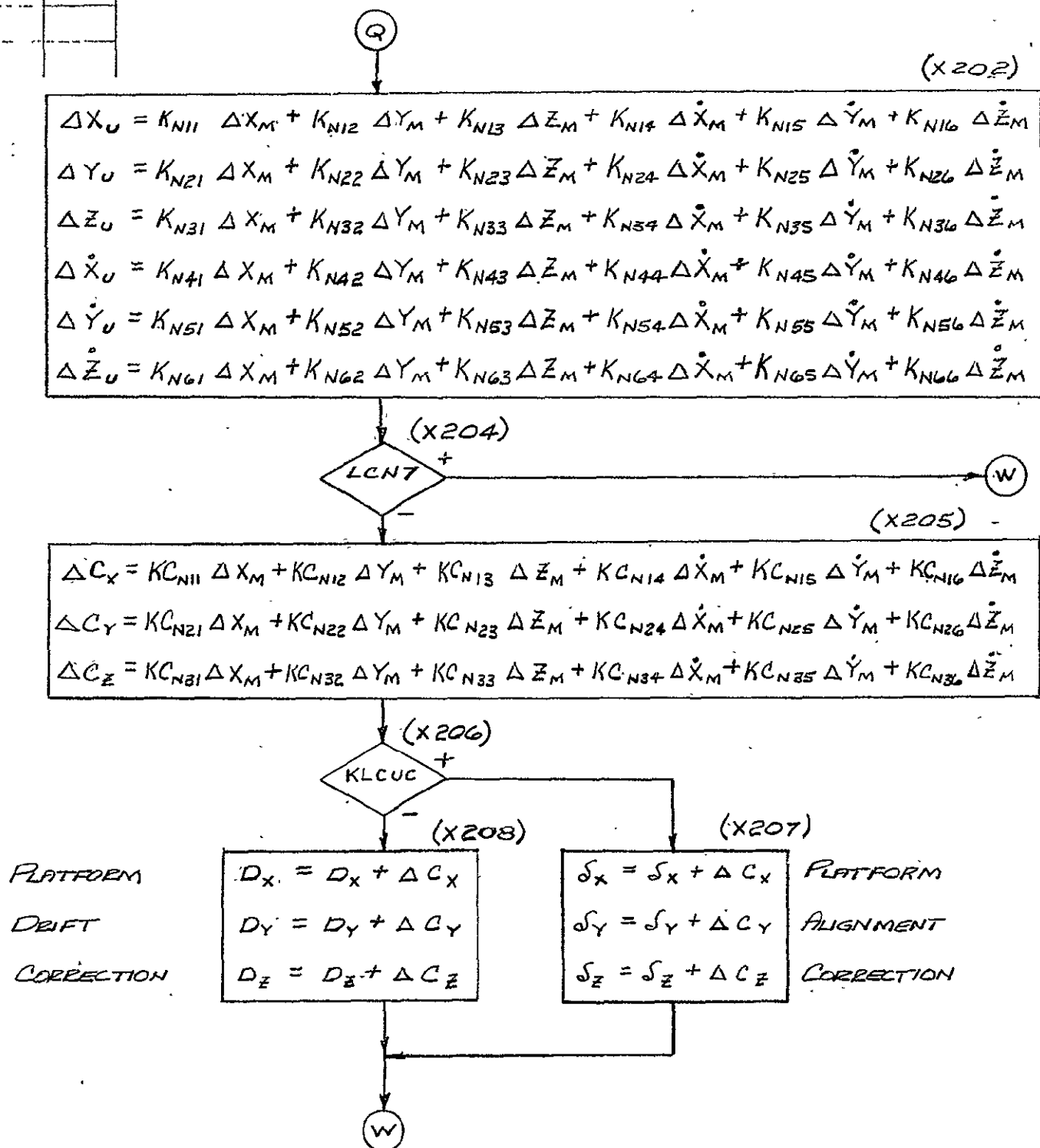
PREPROCESSOR FILTER

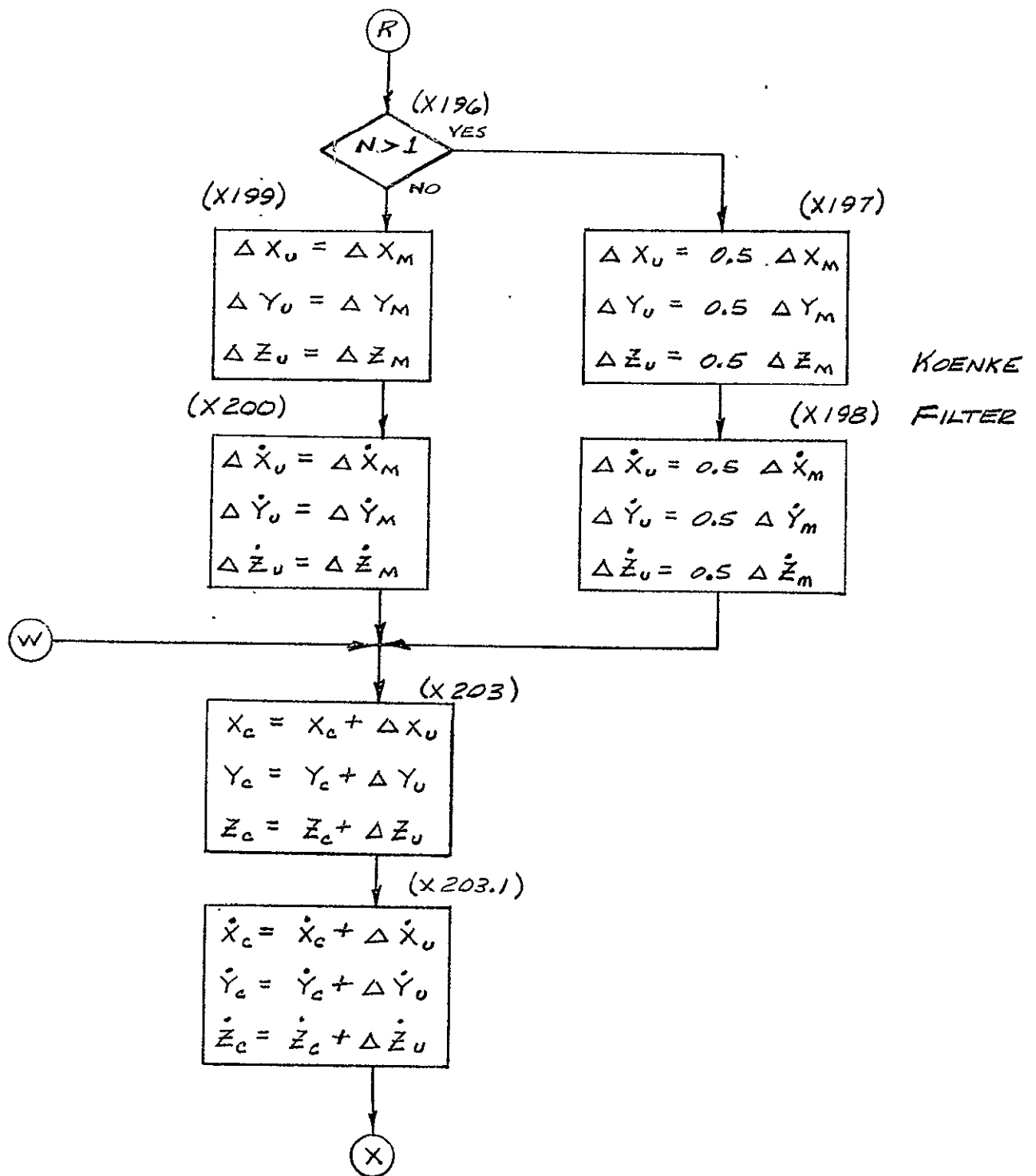


9-23-68	95

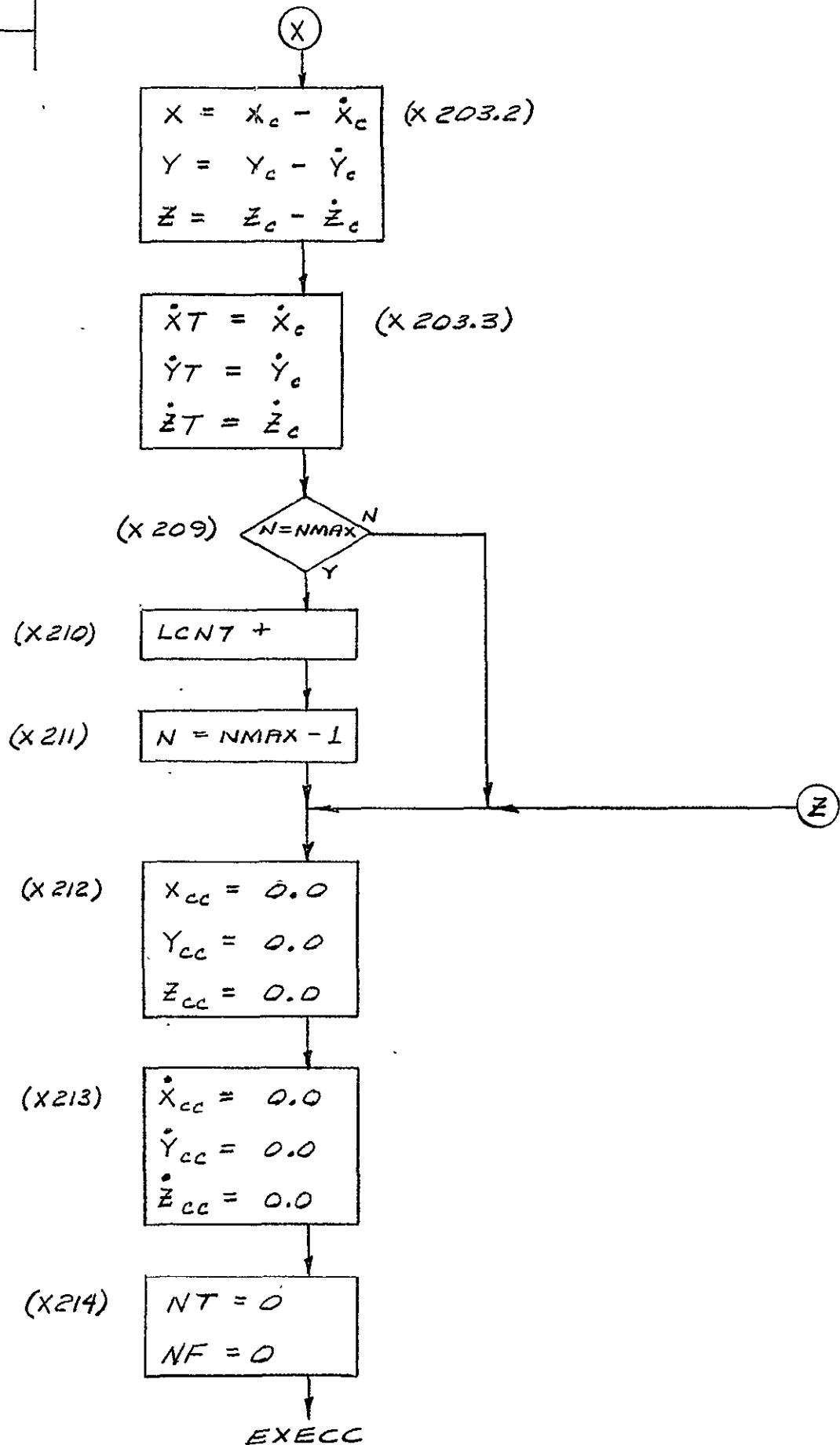


DATE	No.
9-23-68	95

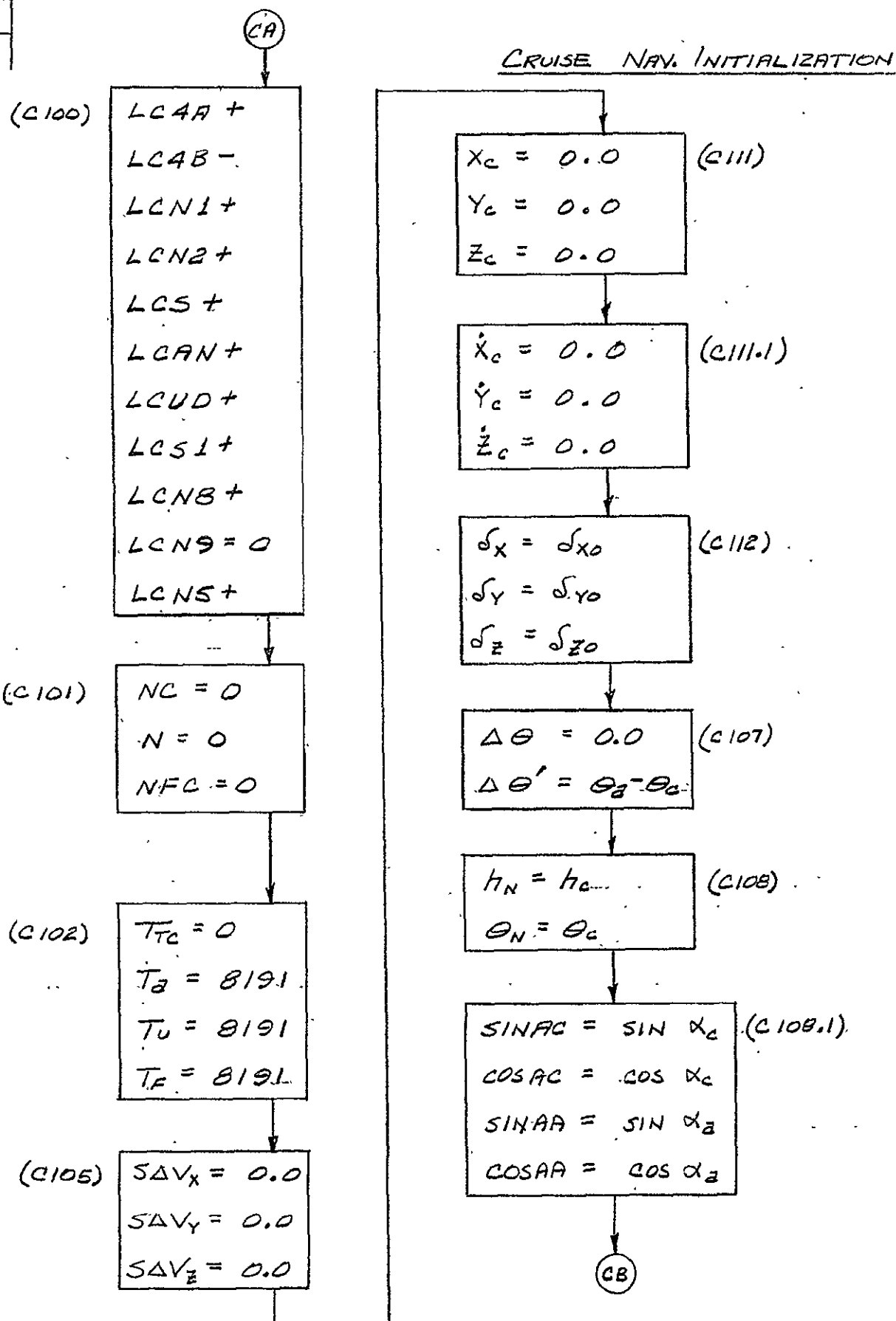




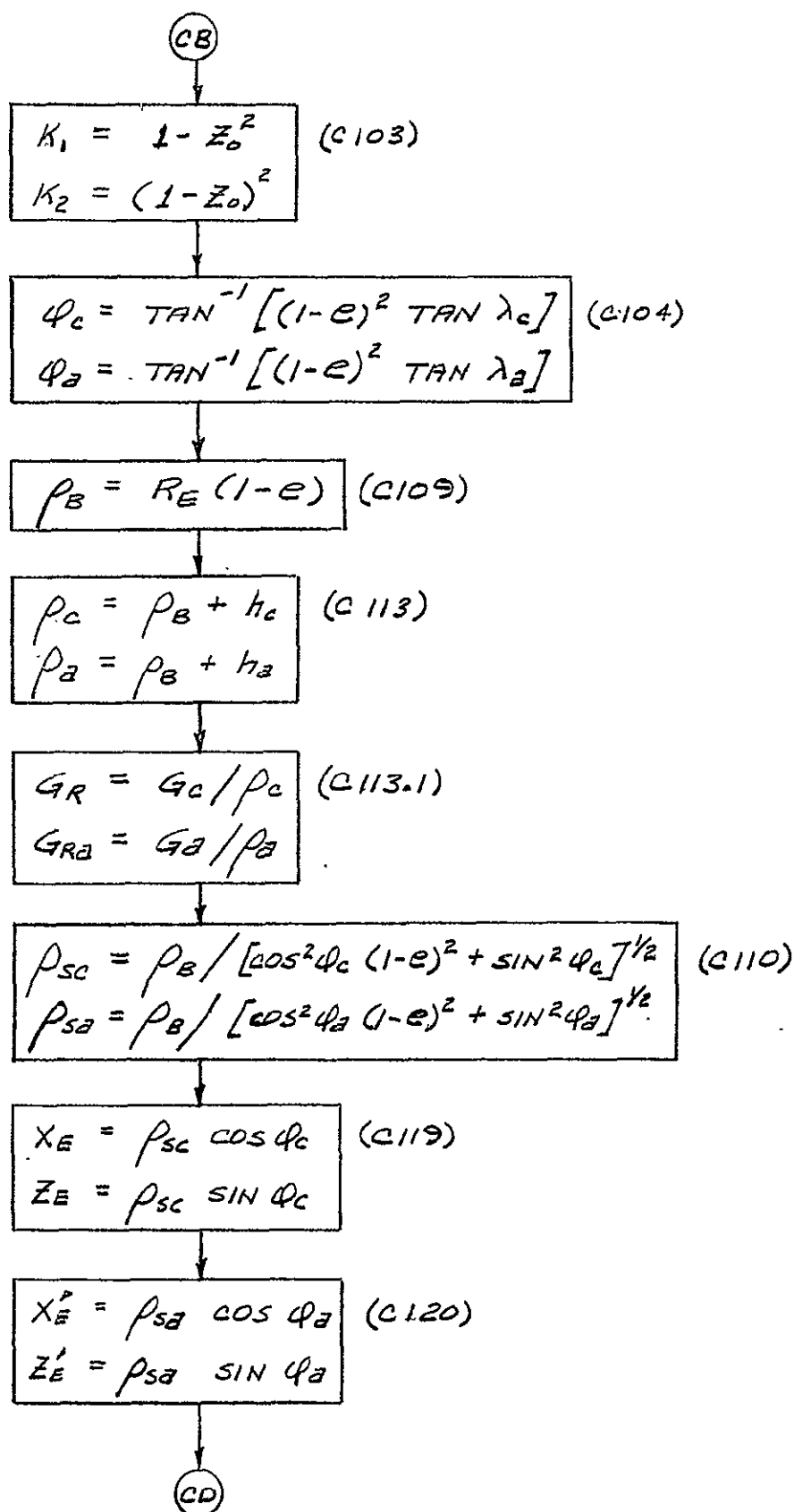
9-23-68	95
10-7-68	100



DATE	No.
9-23-68	95



DATE	No.
9-23-68	95



CD

(C114)

$$D_{11} = -\sin \lambda_c \sin \alpha_c$$

$$D_{12} = \cos \alpha_c$$

$$D_{13} = \cos \lambda_c \sin \alpha_c$$

$$D_{21} = -\sin \lambda_c \cos \alpha_c$$

$$D_{22} = -\sin \alpha_c$$

$$D_{23} = \cos \lambda_c \cos \alpha_c$$

$$D_{31} = \cos \lambda_c$$

$$D_{33} = \sin \lambda_c$$

(C115)

$$C_{12} = -2.0 \omega_e \sin \lambda_c$$

$$C_{13} = 2.0 \omega_e \cos \lambda_c \cos \alpha_c$$

$$C_{21} = -C_{12}$$

$$C_{23} = -2.0 \omega_e \cos \lambda_c \sin \alpha_c$$

$$C_{31} = -C_{13}$$

$$C_{32} = -C_{23}$$

CE

DATE	No.
9-23-68	95

CE

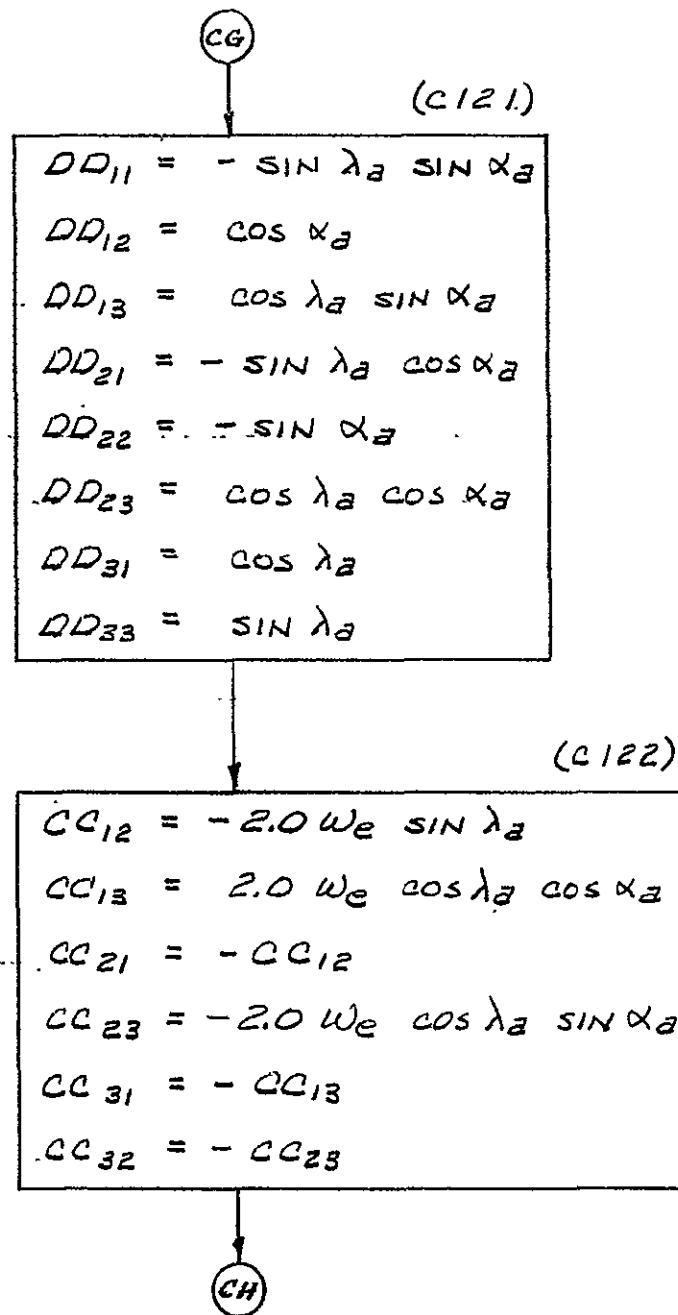
(C116)

$$\begin{aligned}
 A_{11} &= \omega_e^2 (\cos^2 \lambda_c \sin^2 \alpha_c - 1.0) \\
 A_{12} &= \omega_e^2 \cos^2 \lambda_c \cos \alpha_c \sin \alpha_c \\
 A_{13} &= \omega_e^2 \cos \lambda_c \sin \lambda_c \sin \alpha_c \\
 A_{21} &= A_{12} \\
 A_{22} &= \omega_e^2 (\cos^2 \lambda_c \cos^2 \alpha_c - 1.0) \\
 A_{23} &= \omega_e^2 \cos \lambda_c \sin \lambda_c \cos \alpha_c \\
 A_{31} &= A_{13} \\
 A_{32} &= A_{23} \\
 A_{33} &= -\omega_e^2 \cos^2 \lambda_c
 \end{aligned}$$

(C118)

$$\begin{aligned}
 P_{11} &= \sin \lambda_c \sin \alpha_p \\
 P_{12} &= -\cos \lambda_c \\
 P_{13} &= -\sin \lambda_c \cos \alpha_p \\
 P_{21} &= \cos \alpha_p \\
 P_{23} &= \sin \alpha_p \\
 P_{31} &= -\cos \lambda_c \sin \alpha_p \\
 P_{32} &= -\sin \lambda_c \\
 P_{33} &= \cos \lambda_c \cos \alpha_p
 \end{aligned}$$

CG



DATE	NO.
9-23-68	95

(CH)

$$\begin{aligned}
 AA_{11} &= \omega_e^2 (\cos^2 \lambda_a \sin^2 \alpha_a - 1) & (C123) \\
 AA_{12} &= \omega_e^2 \cos^2 \lambda_a \cos \alpha_a \sin \alpha_a \\
 AA_{13} &= \omega_e^2 \cos \lambda_a \sin \lambda_a \sin \alpha_a \\
 AA_{21} &= AA_{12} \\
 AA_{22} &= \omega_e^2 (\cos^2 \lambda_a \cos^2 \alpha_a - 1) \\
 AA_{23} &= \omega_e^2 \cos \lambda_a \sin \lambda_a \cos \alpha_a \\
 AA_{31} &= AA_{13} \\
 AA_{32} &= AA_{23} \\
 AA_{33} &= -\omega_e^2 \cos^2 \lambda_a
 \end{aligned}$$

$$\begin{aligned}
 \ddot{\rho}_{xct} &= \omega_e^2 \rho_{sc} \sin \lambda_c \cos \varphi_c \sin \alpha_c + h_c AA_{13} & (C126) \\
 \ddot{\rho}_{yct} &= \omega_e^2 \rho_{sc} \sin \lambda_c \cos \varphi_c \cos \alpha_c + h_c AA_{23} \\
 \ddot{\rho}_{zct} &= -\omega_e^2 \rho_{sc} \cos \lambda_c \cos \varphi_c + h_c AA_{33}
 \end{aligned}$$

$$\begin{aligned}
 \ddot{\rho}'_{xct} &= \omega_e^2 \rho_{sa} \sin \lambda_a \cos \varphi_a \sin \alpha_a + h_a AA_{13} & (C128) \\
 \ddot{\rho}'_{yct} &= \omega_e^2 \rho_{sa} \sin \lambda_a \cos \varphi_a \cos \alpha_a + h_a AA_{23} \\
 \ddot{\rho}'_{zct} &= -\omega_e^2 \rho_{sa} \cos \lambda_a \cos \varphi_a + h_a AA_{33}
 \end{aligned}$$

$$\begin{aligned}
 G_{x0} &= -G_c \sin \alpha_c \sin \epsilon_{gc} & (C129) \\
 G_{y0} &= -G_c \cos \alpha_c \sin \epsilon_{gc} \\
 G_{z0} &= G_c \cos \epsilon_{gc}
 \end{aligned}$$

(CK)

CK

(C130)

$$\begin{aligned}G_{xa} &= -G_a \sin \alpha_a \sin \epsilon_{ga} \\G_{ya} &= -G_a \cos \alpha_a \sin \epsilon_{ga} \\G_{za} &= G_a \cos \epsilon_{ga}\end{aligned}$$

(C132)

$$\begin{aligned}Q_{11} &= \cos \Delta\theta' \\Q_{12} &= \sin \Delta\theta' \\Q_{21} &= -Q_{12} \\Q_{22} &= Q_{11}\end{aligned}$$

(C133)

$$\begin{aligned}\omega'_{11} &= DD_{11} Q_{11} + DD_{12} Q_{21} \\\omega'_{12} &= DD_{11} Q_{12} + DD_{12} Q_{22} \\\omega'_{13} &= DD_{13} \\\omega'_{21} &= DD_{21} Q_{11} + DD_{22} Q_{21} \\\omega'_{22} &= DD_{21} Q_{12} + DD_{22} Q_{22} \\\omega'_{23} &= DD_{23} \\\omega'_{31} &= DD_{31} Q_{11} \\\omega'_{32} &= DD_{31} Q_{12} \\\omega'_{33} &= DD_{33}\end{aligned}$$

CL

(CL)

C134

$$\begin{aligned}\omega_{11} &= \omega'_{11} D_{11} + \omega'_{12} D_{12} + \omega'_{13} D_{13} \\ \omega_{12} &= \omega'_{11} D_{21} + \omega'_{12} D_{22} + \omega'_{13} D_{23} \\ \omega_{13} &= \omega'_{11} D_{31} + \omega'_{13} D_{33} \\ \omega_{21} &= \omega'_{21} D_{11} + \omega'_{22} D_{12} + \omega'_{23} D_{13} \\ \omega_{22} &= \omega'_{21} D_{21} + \omega'_{22} D_{22} + \omega'_{23} D_{23} \\ \omega_{23} &= \omega'_{21} D_{31} + \omega'_{23} D_{33} \\ \omega_{31} &= \omega'_{31} D_{11} + \omega'_{32} D_{12} + \omega'_{33} D_{13} \\ \omega_{32} &= \omega'_{31} D_{21} + \omega'_{32} D_{22} + \omega'_{33} D_{23} \\ \omega_{33} &= \omega'_{31} D_{31} + \omega'_{33} D_{33}\end{aligned}$$

C135

$$\begin{aligned}Q_X &= X_E Q_{11} - X'_E \\ Q_Y &= X_E Q_{21} \\ Q_Z &= Z_E - Z'_E\end{aligned}$$

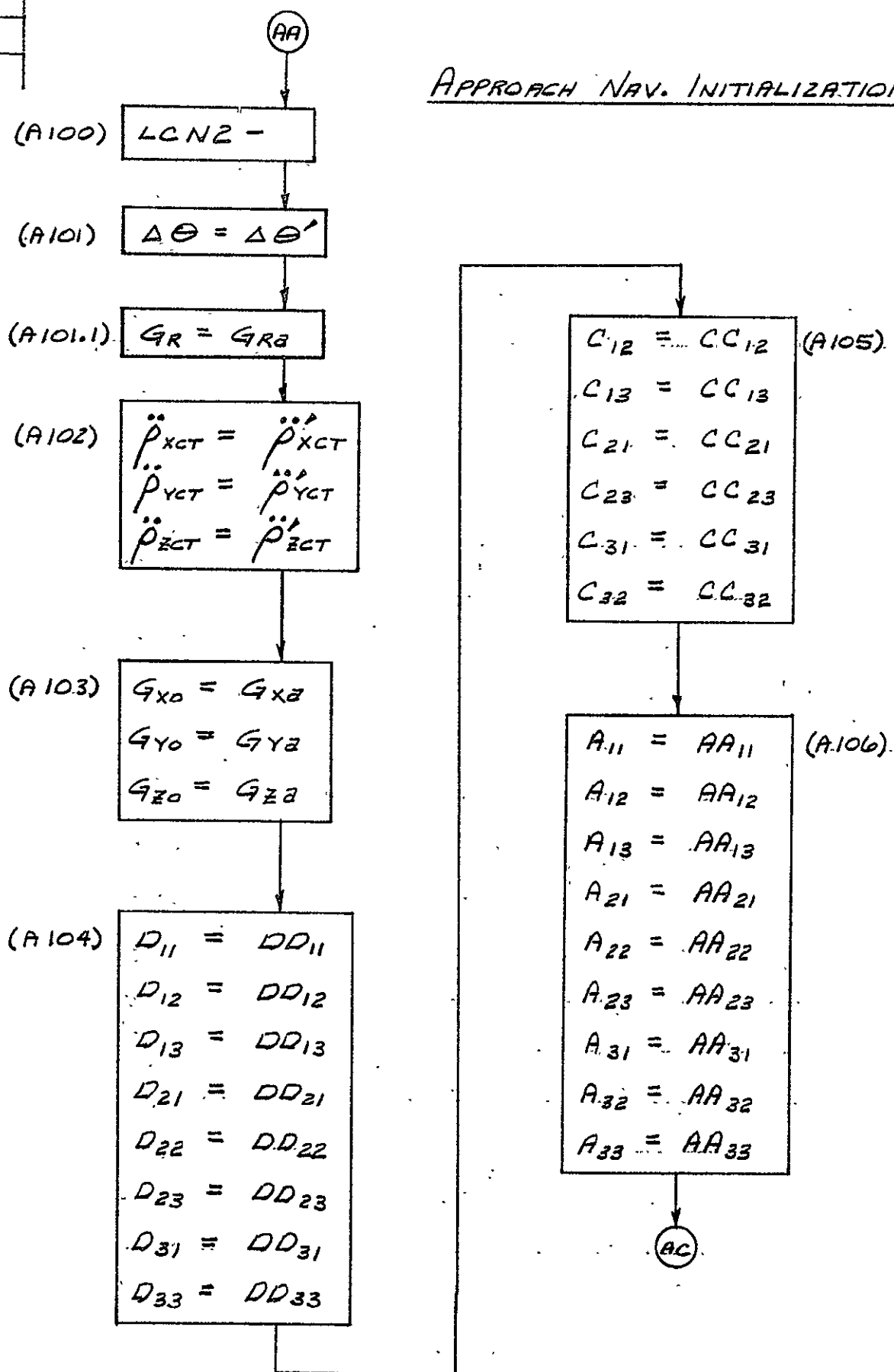
C136

$$\begin{aligned}\rho_X &= DD_{11} Q_X + DD_{12} Q_Y + DD_{13} Q_Z \\ \rho_Y &= DD_{21} Q_X + DD_{22} Q_Y + DD_{23} Q_Z \\ \rho_Z &= DD_{31} Q_X + DD_{33} Q_Z - h_a\end{aligned}$$

EXECIN

DATE	No.
9-23-68	95

APPROACH NAV. INITIALIZATION



DATE	No.
9-23-68	95

AC

$$\begin{aligned} X_a &= w_{11} X_c + w_{12} Y_c + w_{13} (Z_c + h_c) + \rho_x \quad (A108) \\ Y_a &= w_{21} X_c + w_{22} Y_c + w_{23} (Z_c + h_c) + \rho_y \\ Z_a &= w_{31} X_c + w_{32} Y_c + w_{33} (Z_c + h_c) + \rho_z \end{aligned}$$

$$\begin{aligned} X_c &= X_a \quad (A109) \\ Y_c &= Y_a \\ Z_c &= Z_a \end{aligned}$$

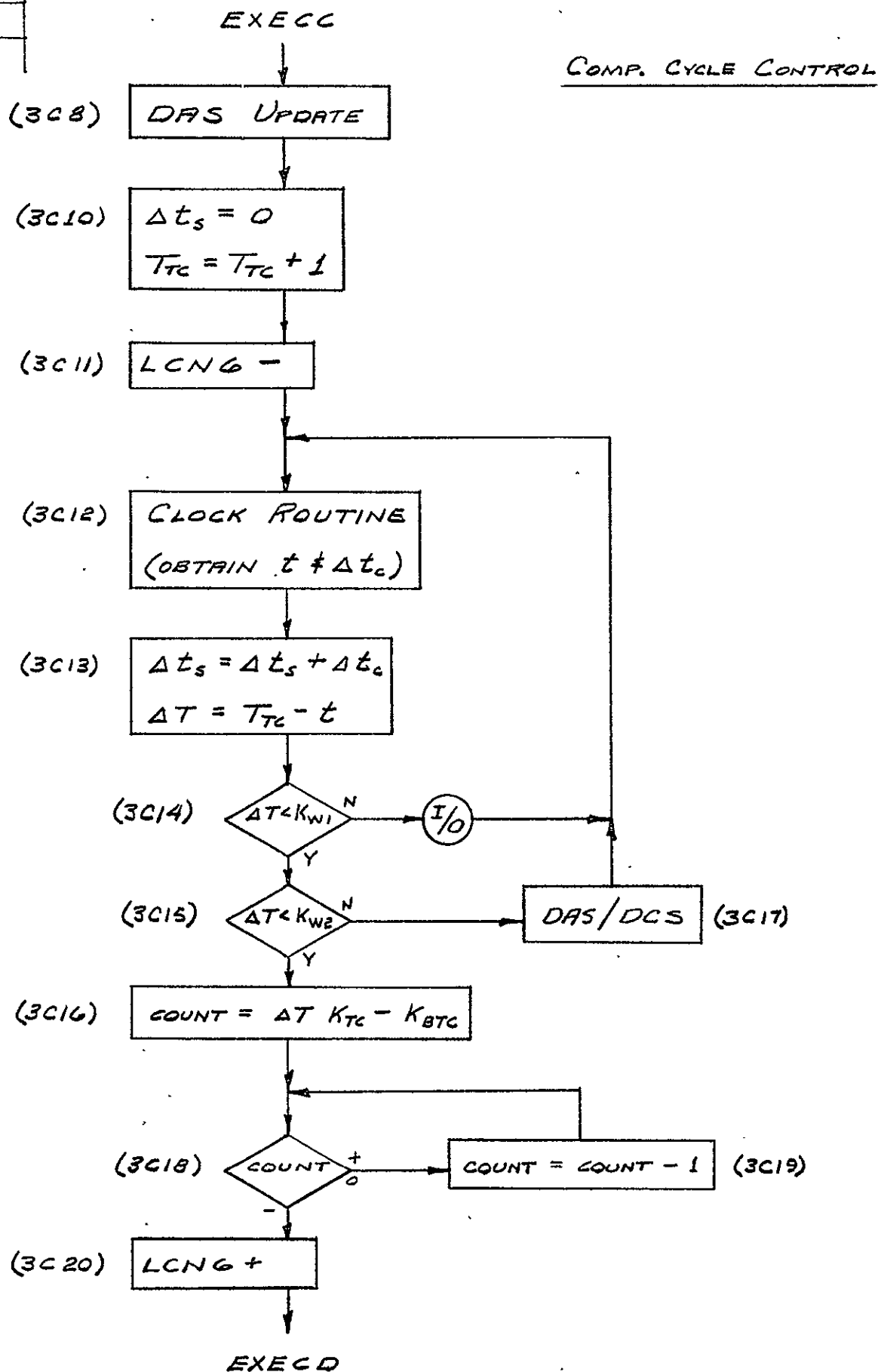
$$\begin{aligned} \dot{X}_a &= w_{11} \dot{X}_c + w_{12} \dot{Y}_c + w_{13} \dot{Z}_c \quad (A110) \\ \dot{Y}_a &= w_{21} \dot{X}_c + w_{22} \dot{Y}_c + w_{23} \dot{Z}_c \\ \dot{Z}_a &= w_{31} \dot{X}_c + w_{32} \dot{Y}_c + w_{33} \dot{Z}_c \end{aligned}$$

$$\begin{aligned} \dot{X}_c &= \dot{X}_a \quad (A111) \\ \dot{Y}_c &= \dot{Y}_a \\ \dot{Z}_c &= \dot{Z}_a \end{aligned}$$

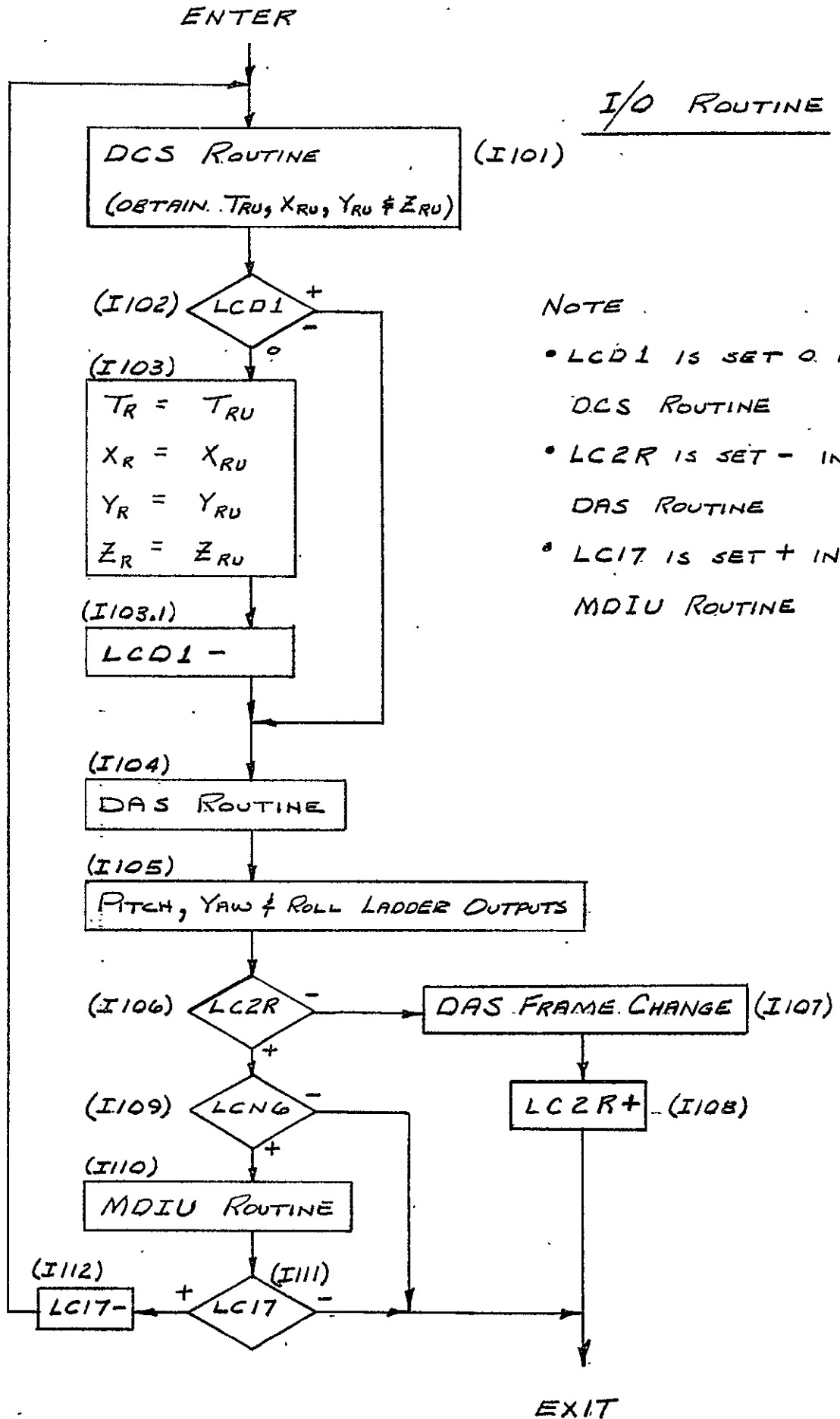
$$\begin{aligned} h_N &= h_a \quad (A114) \\ \Theta_N &= \Theta_a \\ X_E &= X'_E \\ Z_E &= Z'_E \\ \cos AC &= \cos AA \\ \sin AC &= \sin AA \end{aligned}$$

AE

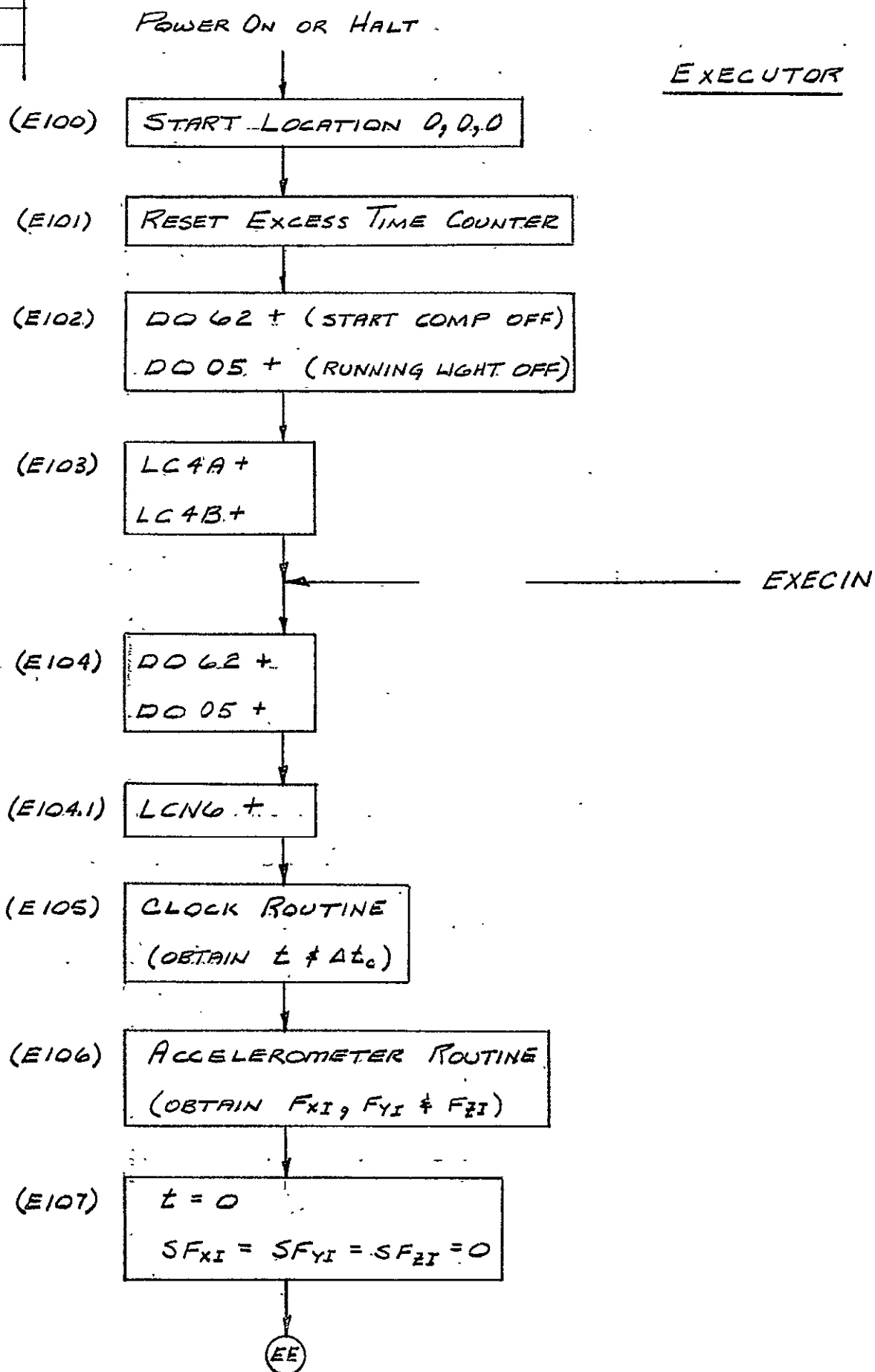
DATE	No.
9-23-68	95



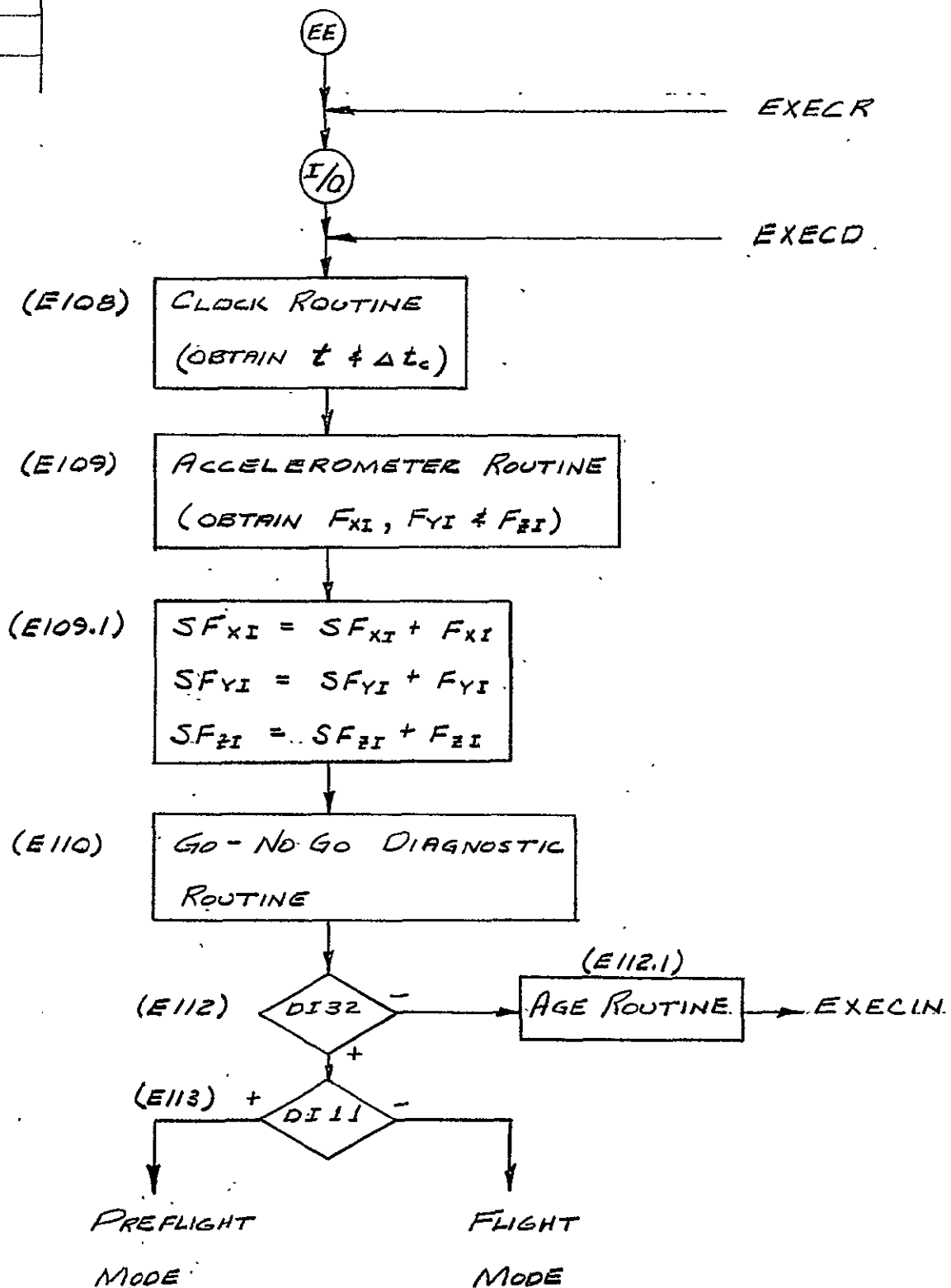
DATE	No.
9-23-68	95



DATE	No.
9-23-68	95



DATE	NO.
9-23-68	95



SYSTEM MATH FLOW SYMBOLS

Symbol	Definition
DELAY	Time delay to reduce syn error between radar and computer clocks
e	Ellipticity of earth
G_a	Magnitude of gravity in approach frame
G_c	Magnitude of gravity in cruise frame
h_a	Height of approach frame origin above reference geoid
h_c	Height of cruise frame origin above reference geoid
K_1	Preprocessor filter gain
K_2	Preprocessor filter gain
K_N	Bad data test logic (for self test)
K_{NT}	Bad data test logic (for self test)
N	Indicates which set of filter weights to use
NMAX	The number of sets of filter weights available
NC	The total number of updates completed
NCFM	Number of rejected radar measurements allowed before restarting update filter
NF	The number of rejected radar measurements since the last initialization

Symbol	Definition
NFC	The total number of rejected radar measurements
NFMAX	The number of rejected radar measurements allowed before re-initialization if forced
NP	Counter used in preprocessor initialization
NSTS	Mode status indicator: 0 initialization; 1 cruise nav. ; 2 approach nav. ; 3 update in cruise; 4 update in approach
NT	The number of passes through the preprocessor since initialization or update
NTMAX	The number of preprocessor passes per update
R_e	Earth equatorial radius
R_r	Norm of radar position vector
Z_0	Preprocessor filter coefficient
t_c	Time since "start comp" (i. e. , total navigation time)
Δt_c	Integration interval
Δt_2	Half the integration interval
Δt_s	Time saved from fixed comp cycle control
T_a	Time to begin approach navigation
T_F	Time to inhibit radar updates
T_R	Time of radar sample
T_S	Self test stop time
T_{TC}	Integer time counter for fixed comp cycled control

Symbol	Definition
\dot{T}_u	Time to begin radar update
V_{ET}	Vehicle east velocity
V_{NH}	Vehicle north velocity
K_{x4}, K_{y4}, K_{z4}	Accelerometer bias (Quanta)
$(K_x, K_y, K_z)_{1,2,3}$	Accelerometer scale factor and misalignment coefficient
$\Delta C_x, \Delta C_y, \Delta C_z$	Differential correction to platform drift or alignment from update filter
D_x, D_y, D_z	Gyro fixed drift rate
F_x, F_y, F_z	Accelerometer output in ft. /sec. (corrected for platform misalignment and bias)
F'_x, F'_y, F'_z	Accelerometer output with bias correction
F_{xI}, F_{yI}, F_{zI}	Accelerometer output (quanta)
F_{xs}, F_{ys}, F_{zs}	Accelerometer output for self test (quanta)
G_{xo}, G_{yo}, G_{zo}	Gravity components at cruise frame origin
G_{xa}, G_{ya}, G_{za}	Gravity components at approach frame origin
R_x, R_y, R_z	Radar data scale factor
$\Delta R, \Delta A, \Delta E$	Radar boresighting residuals
$SF_{XI}, SF_{YI}, SF_{ZI}$	Summation of accelerometer pulses
$S\Delta V_x, S\Delta V_y, S\Delta V_z$	Summation of corrected accelerometer output
U_{IX}, U_{IY}, U_{IZ}	Gyro input axis mass unbalance
U_{sx}, U_{sy}, U_{sz}	Gyro spin axis mass unbalance

Symbol	Definition
X, Y, Z	Position output from radar preprocessor
$\Delta X, \Delta Y, \Delta Z$	Difference between predicted and measured radar position
X_a, Y_a, Z_a	Position in navigation frame at start of approach
X_b, Y_b, Z_b	Radar data bias (Quanta)
X_c, Y_c, Z_c	Position in navigation frame
$\Delta X_c, \Delta Y_c, \Delta Z_c$	Integrated increment of position change
X_d, Y_d, Z_d	Radar measured position synchronized to computer time
X_{cc}, Y_{cc}, Z_{cc}	Second integral of total acceleration since preprocessor initialization or last update
X_{cp}, Y_{cp}, Z_{cp}	Nav. frame position predicted ahead one-half computation cycle
X_E, Y_E, Z_E	Position of cruise frame origin in earth centered fixed frame
X'_E, Y'_E, Z'_E	Position of approach frame origin in earth centered fixed frame
X_{EC}, Y_{EC}, Z_{EC}	Vehicle position in earth centered fixed frame
$\Delta X_m, \Delta Y_m, \Delta Z_m$	Differential between predicted positions and that computed from IMU
X_p, Y_p, Z_p	Predicted position in radar preprocessor
X_R, Y_R, Z_R	Radar measured position (quanta)
X_r, Y_r, Z_r	Corrected radar data minus X_{cc}, Y_{cc}, Z_{cc}
$\Delta X_r, \Delta Y_r, \Delta Z_r$	Radar bore sighting corrections

Symbol	Definition
X_r', Y_r', Z_r'	Radar data with bias and scale factor correction
X_{rc}, Y_{rc}, Z_{rc}	Radar measured position (feet)
$\Delta X_T, \Delta Y_T, \Delta Z_T$	Tolerance for radar data check
$\Delta X_u, \Delta Y_u, \Delta Z_u$	Differential correction to position from update filter
$\dot{X}_c, \dot{Y}_c, \dot{Z}_c$	Velocity in navigation frame
$\Delta \dot{X}_c, \Delta \dot{Y}_c, \Delta \dot{Z}_c$	Total increment of velocity change
$\dot{X}_{cc}, \dot{Y}_{cc}, \dot{Z}_{cc}$	Integral of total acceleration since preprocessor initialization or last update
$\dot{X}_d, \dot{Y}_d, \dot{Z}_d$	Radar measured velocity synchronized to computer time
$\Delta \dot{X}_m, \Delta \dot{Y}_m, \Delta \dot{Z}_m$	Differential between predicted velocity and that computed from IMU
$\Delta \dot{X}_p, \Delta \dot{Y}_p, \Delta \dot{Z}_p$	Corrected accelerometer outputs in platform frame
$\Delta \dot{X}_p', \Delta \dot{Y}_p', \Delta \dot{Z}_p'$	Corrected accelerometer outputs in navigation frame
$\Delta \dot{X}_s, \Delta \dot{Y}_s, \Delta \dot{Z}_s$	Integrated increment of computed velocity
$\dot{X}_T, \dot{Y}_T, \dot{Z}_T$	Velocity output from radar preprocessor
$\Delta \dot{X}_u, \Delta \dot{Y}_u, \Delta \dot{Z}_u$	Differential correction to velocity from update filter
$\ddot{X}_g, \ddot{Y}_g, \ddot{Z}_g$	Computed gravitational acceleration
$\ddot{X}_{CR}, \ddot{Y}_{CR}, \ddot{Z}_{CR}$	Computed coriolis acceleration
$\ddot{X}_{CT}, \ddot{Y}_{CT}, \ddot{Z}_{CT}$	Computed centripetal acceleration

Symbol	Defintion
α_a	Approach frame azimuth
α_c	Cruise frame azimuth
α_p	Platform azimuth
ϵ_{ga}	Angle between plump bob and gravity vector (approach)
ϵ_{gc}	Angle between plump bob and gravity vector (cruise)
θ	Integral of earth rate since start of navigation
θ_a	Longitude of approach frame origin
θ_c	Longitude of cruise frame origin
θ_v	Vehicle longitude
$\Delta\theta$	Constant used in cruise frame computation
$\Delta\theta'$	Longitude between cruise and approach frame origins
λ_a	Geodetic latitude of approach frame origin
λ_c	Geodetic latitude of cruise frame origin
λ_v	Vehicle latitude
ρ_a	Geodetic radius of approach frame origin
ρ_c	Geodetic radius of cruise frame origin
ρ_B	Mean radius of earth's polar axis
ρ_{sa}	Geocentric radius to reference geoid for approach frame

Symbol	Definition
ρ_{sc}	Geocentric radius to reference geoid for cruise frame
ϕ_a	Geocentric latitude of approach frame origin
ϕ_c	Geocentric latitude of cruise frame origin
ω_e	Earth rotational rate
$\delta_{xo}, \delta_{yo}, \delta_{zo}$	Initial platform misalignment
$\delta_x, \delta_y, \delta_z$	Computed platform misalignment
ρ_x, ρ_y, ρ_z	Position components used in approach initialization
$\ddot{\rho}_{XCT}, \ddot{\rho}_{YCT}, \ddot{\rho}_{ZCT}$	Centripetal acceleration of cruise frame origin
$\ddot{\rho}'_{XCT}, \ddot{\rho}'_{YCT}, \ddot{\rho}'_{ZCT}$	Centripetal acceleration of approach frame origin

Logic Choices

Symbol	Definition
LC4A	Logic choice to control preflight mode: + allows preflight initialization; - implies standard preflight
LC4B	Logic choice to control cruise navigation initialization: + allows cruise initialization; - allows navigation
LCAN	Logic choice to force the approach navigation mode: + allows cruise navigation; - forces approach navigation (DCS)
LCUD	Logic choice (3 way) to control radar update: + inhibits update; 0 forces update; - forces update initialization (DCS)
KLCUC	Logic choice to determine how the platform corrections are used: + corrects for platform alignment; - corrects for platform drift
LCN1	Logic choice (3 ways) to control initial program sequencing: + delays navigation until "start comp"; 0 zeros summed accelerations and time; - allows execution of the navigation equations
LCN2	Logic choice to indicate navigation frame: + implies cruise navigation and allows test to begin approach; - implies approach navigation
LCN3	Logic choice to indicate which filter weights to use: + use Koenke filter; - implies use of prestored weights

Symbol	Definition
LCN4	Logic choice indicating status of preprocessor filter initialization: + implies initialization not complete; - implies initialization completion
LCN5	Logic choice indicating acceptability of radar measured data: + implies data can be used in preprocessor; - implies data unacceptable for preprocessor
LCN6	Logic choice to inhibit MDIU: + allows MDIU; - inhibits MDIU
LCN7	Logic choice to determine if platform correction is computed: + implies no platform correction; - implies platform correction
LCS	Logic choice to execute self test mode: + implies regular navigation mode; 0 or - implies test modes
LCS1	Logic choice for self test mode: + implies regular navigation mode; - implies test mode
KLCUC	Logic choice to determine if platform drift or misalignment is corrected when using nine state mixing filter; + implies alignment correction; - implies drift correction
DI13	Input discreet for pilot override on radar update
DI21	Input discreet from "start comp" button
DO05	Output discreet used to turn running light on and free platform

Matrices

Symbol	Definition
A	Transforms navigation position to centripetal acceleration (cruise)
AA	Transforms navigation position to centripetal acceleration (approach)
C	Transforms navigation velocity to coriolis acceleration (cruise)
CC	Transforms navigation velocity to coriolis acceleration (approach)
D	Transforms ECF to ESF (cruise)
DD	Transforms ECF to ESF (approach)
N	Transforms ECI to ECF
P	Transforms platform to ECI
Q	Transforms ECF (cruise) to ECF (approach)
W	Transforms ESF (cruise) to ESF (approach)
K_{NIJ}	Prestored filter weights for position and velocity updates
KC_{NIJ}	Prestored filter weights for platform corrections

Appendix D

MDIU/DCS/DAS QUANTITIES

The input data required by the Gemini computer program is contained in Tables D-1 and D-2. Those quantities in Table D-1 are entered via the telemetry up-link, i. e. , the digital command system (DCS); and those in Table D-2 are inserted via the manual data insertion unit (MDIU) prior to flight.

Tables D-3 and D-4 list the quantities that are telemetered down via the digital acquisition system (DAS) and recorded for post flight analysis.

Table D-1 DCS Quantities

Address	Symbol	Definition	Input			Computer		Comments
			Format	Range	Units	Scaling	Units	
00		Spare						
01		"						
02		"						
03		"						
04		"						
05		"						
06		"						
07		"						
08		"						
09		"						
10	LCUD	Update control logic choice	B 25	+ or -	ND	B 25	ND	- Initiate update + Inhibit update
11	LCAN	Logical choice for approach navigation	B 25	+ or -	ND	B 25	ND	- Switch to approach nav.
12	T _{ru}	Radar data time	B 13	0 to 8191	Sec.	B 13	Sec.	High order bit in normal DCS sign position
13	X _{ru}	Radar Position data	B 13	0 to \pm 8191	Quanta	B 13	Quanta	
14	Y _{ru}	"	"	"	"	"	"	
15	Z _{ru}	"	"	"	"	"	"	

Table D-2 MDIU Quantities

Address	Symbol	Definition	Input			Computer		Comments
			Format	Range	Units	Scaling	Units	
16	θ_c	Longitude of cruise frame origin	xxx. xx	0 to 360	Deg.	B 3	Rad.	
17	λ_c	Latitude of cruise frame origin	+xx. xx	0 to + 67	Deg.	B 3	Rad.	
18	h_c	Height of cruise frame origin above reference spheroid	+xxxx.	0 to 500	Yards	B 24	Ft.	
19	θ_a	Longitude of approach frame origin	xxx. xx	0 to 360	Deg.	B 3	Rad.	
20	λ_a	Latitude of approach frame origin	+xx. xx	0 to + 67	Deg.	B 3	Rad.	
21	h_a	Height of approach frame origin above reference spheroid	+xxxx.	0 to 500	Yards	B 24	Ft.	
22	ΔX_T	Radar data tolerance	xxxx. x	0 to 1000	Ft.	B 20	Ft.	
23	ΔY_T	Radar data tolerance	xxxx. x	0 to 1000	Ft.	B 20	Ft.	
24	ΔZ_T	Radar data tolerance	xxxx. x	0 to 1000	Ft.	B 20	Ft.	
25	LCN7	Logical choice for six or nine state filter	+xxxx	+ or -	ND	B 25	ND	+ six state - nine state
26	NFMAX	Max. no. of bad radar data allowed per update	+xxxx.	0 to 100	ND	B 25	ND	
27	NTMAX	Number of preprocessor cycles per update	+xxxx.	0 to 100	ND	B 25	ND	
28	NMAX	Number of update filter gain sets	+xxxx.	0 to 10	ND	B 25	ND	

Table D-2 MDIU Quantities (Continued)

Address	Symbol	Definition	Input			Computer		Comments
			Format	Range	Units	Scaling	Units	
29	T_u	Time to enable update	xxxx. x	0 to 8191	Sec.	B 13	Sec.	
30	K_{x1}	Accelerometer scale factor	+ .xxxx	0 to +. 24	fps/ pulse	B -2	fps/pulse	
31	K_{x2}	Accelerometer misalign- ment coefficient	"	"	"	"	"	
32	K_{x3}	Accelerometer misalign- ment coefficient	"	"	"	"	"	
33	K_{y1}	Accelerometer misalign- ment coefficient	"	"	"	"	"	
34	K_{y2}	Accelerometer scale factor	"	"	"	"	"	
35	K_{y3}	Accelerometer misalign- ment coefficient	"	"	"	"	"	
36	K_{z1}	Accelerometer misalign- ment coefficient	"	"	"	"	"	
37	K_{z2}	Accelerometer misalign- ment coefficient	"	"	"	"	"	
38	K_{z3}	Accelerometer scale factor	"	"	"	"	"	
39	K_{x4}	Accelerometer bias coefficient	+x. xxx	0 to + 3.9	pulses/ sec.	B 2	pulses/ sec.	
40	X_{y4}	Accelerometer bias coefficient	+x. xxx	0 to + 3.9	pulses/ sec.	B 2	pulses/ sec.	
41	K_{z4}	Accelerometer bias coefficient	+x. xxx	0 to + 3.9	pulses/ sec.	B 2	pulses/ sec.	

Table D-2 MDIU Quantities (Continued)

Address	Symbol	Definition	Input			Computer		Comments
			Format	Range	Units	Scaling	Units	
42	ΔR	Radar range bias	$\pm xxx. x$	0 to ± 500	Ft.	B 13	Ft.	
43	ΔA	Radar azimuth misalignment	$\pm . xxxxx$	0 to $\pm .9$	Deg.	B 3	Rad.	
44	ΔE	Radar elevation misalignment	$\pm . xxxxx$	0 to $\pm .9$	Deg.	B 3	Rad.	
45	δ_{xo}	Initial Platform misalignment	$\pm x. xxx$	0 to ± 1.75	Deg.	B -5	Rad.	
46	δ_{yo}	Initial platform misalignment	"	"	"	"	"	
47	δ_{zo}	Initial platform misalignment	"	"	"	"	"	
48		Spare						
49		Spare						
50	D_x	Gyro fixed drift rate	$\pm x. xxx$	0 to ± 2	Deg/ Hr	B -10	Rad/sec	
51	D_y	Gyro fixed drift rate	"	"	"	"	"	
52	D_z	Gyro fixed drift rate	"	"	"	"	"	
53	U_{sx}	Gyro spin axis mass unbalance coefficient	$\pm x. xxx$	0 to ± 1	$\frac{\text{Deg/hr}}{G}$	B -17	$\frac{\text{Rad/sec}}{\text{Ft/sec}^2}$	
54	U_{xy}	"	"	"	"	"	"	
55	U_{sz}	"	"	"	"	"	"	
56	U_{Ix}	Gyro input axis mass unbalance coefficient	"	"	"	"	"	

Table D-2 MDIU Quantities (Continued)

Address	Symbol	Definition	Input			Computer		Comments
			Format	Range	Units	Scaling	Units	
57	U_{ly}	Gyro input axis mass unbalance coefficient	$\pm x. xxx$	0 to ± 1	$\frac{\text{Deg/hr}}{G}$	B -17	$\frac{\text{Rad/sec}}{\text{Ft/sec}^2}$	
58	U_{lz}	"	"	"	"	"	"	
59		Spare						
60	p	Platform azimuth	xxx. xx	0 to 360	Deg.	B 3	Rad.	
61	c	Cruise frame azimuth	"	"	"	"	"	
62	a	Approach frame azimuth	"	"	"	"	"	
63		Spare						
64		Spare						
65	G_c	Gravity magnitude for cruise frame	xx. xxx	32 ± 0.7	$\frac{\text{Ft.}}{\text{sec.}^2}$	B 6	Ft/sec^2	
66	G_a	Gravity magnitude for approach frame	"	"	"	"	"	
67	ϵ_{gc}	Angle between plumb bob and gravity for cruise	xx. xxx	0 to 10	Min	B 3	Rad.	
68	ϵ_{ga}	Angle between plumb bob and gravity for approach	"	"	"	"	"	
69	NFCM	Max. no. of bad radar allowed between updates before restarting update filter	xxxxx	0 to 100	ND	B 25	ND	
70	T_a	Time to initiate approach navigation	xxxx. x	0 to 8191	Sec.	B 13	Sec.	

Table D-2 MDIU Quantities (Continued)

Address	Symbol	Definition	Input			Computer		Comments
			Format	Range	Units	Scaling	Units	
71	R _x	Radar data scale factor	<u>+</u> x. xxx	<u>+</u> .5 to <u>+</u> 4	<u>Ft</u> <u>Quanta</u>	B 7	<u>Ft</u> <u>Quanta</u>	
72	R _y	"	"	"	"	"	"	
73	R _z	"	"	"	"	"	"	
74	K _{w1}	Comp. cycle control constant	xx. xxx	0 to .1	Sec.	B 13	Sec.	
75	K _{w2}	Comp. cycle control constant	xx. xxx	0 to .2	Sec.	B 13	Sec.	
76	K _{tc}	Comp. cycle control constant	xxxxx.	1000 to 3500	<u>Quanta</u> <u>Sec.</u>	B 12	<u>Quanta</u> <u>Sec.</u>	
77	K _{btc}	Comp. cycle control constant	xxxxx.	1 to 30	Quanta	B 25	Quanta	
78		Spare						
79		Spare.						
80		Not available						
81		"						
82		"						
83	X _b	Radar bias coefficient	<u>+</u> xxx. x	0 to <u>+</u> 8191	Quanta	B 13	Quanta	
84	Y _b	"	"	"	"	"	"	
85	Z _b	"	"	"	"	"	"	
86	LCS1	Self test logic choice	<u>+</u> xxxx	+ or -	ND	B 25	ND	
87	LCS	Self test logic choice	<u>+</u> xxxx	+, - or 0	ND	B 25	ND	

D-7

Table D-2 MDIU Quantities (Continued)

Address	Symbol	Definition	Input			Computer		Comments
			Format	Range	Units	Scaling	Units	
88	T _F	Time to terminate update	xxxx. x	0 to 8191	Sec.	B 13	Sec.	
89		Spare						
90	T _s	Self test stop time	xxxx. x	0 to 8191	Sec.	B 13	Sec.	
91	F _{xs}	Accelerometer output for self test	+xxxx.	0 to + 500	<u>Pulses</u> Sec.	B 12	<u>Pulses</u> Sec.	
92	F _{ys}	"	"	"	"	"	"	
93	F _{zs}	"	"	"	"	"	"	
94	Z _o	Preprocessor filter coefficient	+ .xxxx	.5 to .99	ND	B 1	ND	
95	Delay		+xxxx.	0 to 150	ND	B 25	ND	
96	K _n	Bad data test logic	+xxxx.	0 to 1000	ND	B 25	ND	
97	K _{nt}	Bad data test logic	+xxxx.	0 to 10	ND	B 25	ND	
98	LCN3	Logical choice for update filter	+xxxx	+ or -	ND	B 25	ND	+ Koenke filter - 6 or 9 state filter
99		Spare						

Table D-3 DAS List (Preflight Mode)

Sequence Number	Symbol	Description	Units	Scaling
1	SF_{xi}	Summation of Accelerometer Pulses	Pulses	B 20
2	SF_{yi}	"	"	"
3	SF_{zi}	"	"	"
4	t	Computer Time	Sec.	B 14
5	F_{xi}	Accelerometer Output	Pulses	B 14
6	F_{yi}	"	"	"
7	F_{zi}	"	"	"
8	$\Delta\psi_{bo}$	Ladder Output	Quanta	B 14
9	$\Delta\theta_{bo}$	"	"	"
10	$\Delta\phi_{bo}$	"	"	"
11	θ_b	Platform Gimbal	Quanta	B 14
12	ψ_b	"	"	"
13	ϕ_b	"	"	"
14		Spare		
15		MDIU Address Tag		
16		Multiplexed MDIU Quantities		
17		"		
18		"		
19		"		
20		"		
21		"		

Table D-4 DAS List (Navigation Modes)

Sequence Number	Symbol	Description	Units	Scaling
1	SF_{xi}	Summation of Accelerometer Pulses.	Pulses	B 20
2	SF_{yi}	"	"	"
3	SF_{zi}	"	"	"
4	t	Computer Navigation Time	Sec.	B 14
5	X_c	Position in Navigation Frame	Ft.	B 18
6	Y_c	"	"	"
7	Z_c	"	"	"
8	\dot{X}_c	Velocity in Navigation Frame	Ft. /Sec.	B 11
9	\dot{Y}_c	"	"	"
10	\dot{Z}_c	"	"	"
11	λ_v	Vehicle Latitude	Deg.	B 8
12	θ_v	Vehicle Longitude		B 9
13	V_{NH}	North Velocity	Ft. /Sec.	B 11
14	V_{ET}	East Velocity		
15	ΔX	Delta Position Generated by Preprocessor Filter	Ft.	B 20
16	ΔY	"	"	"
17	ΔZ	"	"	"
*18	T_R , NSTS	Radar Update Time, Mode Status	Sec.	B 14
*19	X_R , NC	Position From Radar, No. of IMU Updates	Quanta	B 14
*20	Y_R , NFC	Position From Radar, No. of Bad Radar Transmissions	"	"
21	Z_R	Position From Radar	"	"

* Radar data contained in first 15 bits with the indicator occupying the remaining 9.